AD-A110 895   NAVAL RESEARCH LAB WASHINGTON DC   F/G 4/1
THE METEOR SOFTWARE PACKAGE FOR ANALYSIS OF METEOROLOGICAL DATA--ETC(U)
JAN 82 J B HOOVER

UNCLASSIFIED   NRL-MR-4674   NL

1 of 1
AD
A110895

END
DATE
FILMED
3-82
DTIC

AD A110895

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NRL Memorandum Report 4674 — 2. GOVT ACCESSION NO.: AD-A ...	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): THE METEOR SOFTWARE PACKAGE FOR ANALYSIS OF METEOROLOGICAL DATA	5. TYPE OF REPORT & PERIOD COVERED: Report on a continuing NRL problem.
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): John B. Hoover	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Research Laboratory, Washington, D.C. 20375	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61153N; 43-1130-0-2
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE: January 27, 1982
	13. NUMBER OF PAGES: 88
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Meteorology
Data acquisition
Data processing
FORTRAN
Computer data processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The METEOR software package was developed at NRL to aid in collecting and analyzing ~~the collection and analysis of~~ meteorological data. It consists of four FORTRAN programs and is currently in use on a DEC System-10. All four programs were designed for ease of use and require a minimal amount of effort on the part of the user.

The programs ~~in this package~~ are capable of sorting data according to source, creating large data files, processing these files, and producing printed and plotted output as required.

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

251950

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued)

—Versatility was a prime objective in creating these programs. They have been written in a modular fashion so that functions may easily be added or changed. In addition, the user may select from a variety of options within each program, thus allowing the software to be tailored to specific requirements at execution time. This is accomplished through the use of a command file which provides instructions to the various programs.

One ~~of the~~ programs is interactive and conducts a dialog with the user in order to ascertain what data is to be processed and which functions are to be performed. The output of this program is the command file described above.

The remaining three programs require no interaction and may be used in a batch mode. Each of these produces error messages whenever unexpected conditions are encountered. Insofar as possible, these messages were intended to be self-explanatory. If more details are required, each program also contains a complete list of error messages and an explanation of each.

This report includes descriptions of the programs, examples of the required inputs, and copies of typical program outputs. Complete source listings are also provided in the appendices.

Acknowledgements

The author would like to thank Katherine Schwarz and Joseph Liu for their assistance in writing programs METCLC and METPLT during their participation in the American University/Naval Research Laboratory Research Apprentice Program.

Elizabeth Hill and Jay Oberfield of the Research Computation Division were instrumental in debugging the programs.

This work was performed while the author was a National Research Council Research Associate at NRL.

Table of Contents

List of Figures and Tables

The METEOR Software Package for Analysis of Meteorological Data

I. INTRODUCTION

A requirement for many experiments in environmental chemistry is that extensive meteorological records must be maintained. Frequently, simultaneous collection of several different types of data (temperature, humidity, pressure, etc.) at regular intervals (minutes to hours) over rather long, continuous periods (days or weeks) is necessary. The use of modern data loggers greatly simplifies the acquisition of data and often provides for storage and transfer on computer-readable media, such as magnetic tape. Offline analysis involves conversion of the data to physically meaningful units, calculation of derived quantities, and presentation of the results in formats which are convenient for the user's purposes. Due to the large quantities of data, these steps are usually very time-consuming.

METEOR is a FORTRAN program package which is designed to alleviate many of these difficulties. It provides software for inputting data files, searching out relevant portions of these files, processing data, and generating printed or plotted output.

The goal has been to maximize the generality of the program while minimizing the demands on the operator. The former goal has been addressed by the liberal use of subroutines and functions, making program expansion and alteration a simple matter of inserting new or updated program modules.

The requirement that the package be easy to use has led to the development of an auxiliary, interactive program which aids in the creation of a command file. This file then controls the execution of the primary programs. Currently, there are three primary programs (METSRT, METCLC, and METPLT) and one auxiliary program (METINP) in the METEOR package. The relationship among these programs is illustrated in Figure 1.

METSRT (METeorological data SoRTing program) is responsible for input of raw data, selection of the data required for analysis, preliminary processing (conversion of units and scaling, where necessary), testing of the data for various error conditions, listing of selected data in tabular form, and creation of a file containing all processed data.

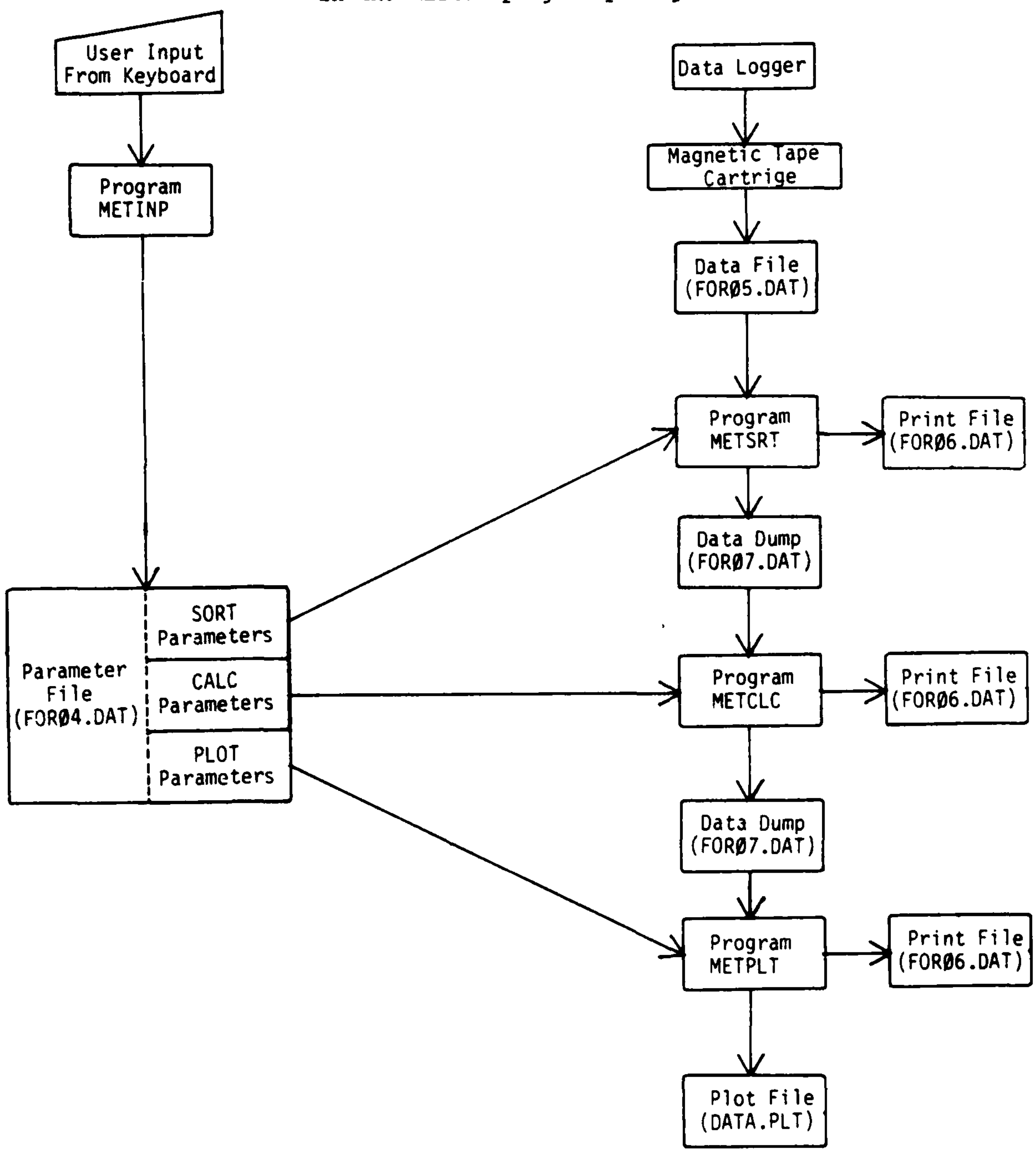
METCLC (METeorological data CaLCulation program) then operates on the output of METSRT, calculating the values of various derived quantities (total moisture loading, for example). These results are added to the file of processed data and may also be printed out, again in tabular form.

METPLT (METeorological data PLoTting program), the last of the primary programs, reads the data file produced by the previous programs and generates selected graphs.

It is possible to string these programs together and to process the data, from raw data input to finished plots, in one continuous batch job. However, the sequential design of the programs, with intermediate outputs, was intended to allow easy operator intervention in the event that bad data is encountered.

Manuscript submitted October 30, 1981.

Figure 1
Relationship of the programs and files
in the METEOR program package.



The auxiliary program, METINP, (METeorological parameter INPut program), is interactive and is intended to be run from a CRT terminal. This program requests information regarding the specific functions in METSRT, METCLC, and METPLT that are desired and constructs a control file in the appropriate format. This file is displayed line-by-line for operator verification before being written onto the disk. Errors may easily be corrected at this point.

The METEOR package is currently running on a DEC System-10 and the I/O file names discussed below are those used by the TOPS-10 operating system. No major problems are expected to arise if the programs are transfered to another system (in fact, earlier versions of METSRT and METPLT were run on a Texas Instruments Advanced Scientific Computer). However, METINP, because it is interactive, will not operate properly in a batch processing environment.

II. METSRT OPERATION

For reference during the following discussion, a complete listing of the METSRT source code is given in Appendix A.

METSRT may be logically divided into input, processing, and output sections. The input, and, to some extent, the processing sections of the program must be tailored to the characteristics of the data source. METSRT was originally written specifically for use with a Fluke Model 2240B data logger, having the analog data output format shown in Table 1. A digital data format (Table 1) is also available, but is not currently in use. Other formats would necessitate changes to the search and input routines and possibly to the error flagging subroutine. We are presently making alterations in order that the output of a newly constructed data acquisition system may be processed by METSRT.

Four I/O files (and four different logical devices) are involved:

1) FORØ4.DAT (device 4) contains parameters which control program execution.
2) FORØ5.DAT (device 5) is the input data file.
3) FORØ6.DAT (device 6) is a tabular output for printing.
4) FORØ7.DAT (device 7) contains all data and parameters and is intended to be read by subsequent programs.

Initially, METSRT reads the control file (FORØ4.DAT), which specifies the dates of interest, the specific types of data which are to be processed, the desired format for printed output, and the units for both input and output. Print switches may be set to select channels for which data is to be listed. In addition, other parameters, pertaining to METCLC and METPLT, may be present. These parameters, if present, are ignored by METSRT. An example of the control file is given in Table 2.

A subroutine (SEARCH) is then called which searches the data file (FORØ5.DAT) for the first data set and reads the time and date header. Any data set which is dated prior to the specified initial time is rejected and the search continues until one of the following conditions is met:

1) An End of File (EOF) is read.

Table 1
Data Logger Format

	Column Number																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Data Set Header	<u>Y</u>	d	d	d	<u>:</u>	h	h	<u>:</u>	m	m	<u>:</u>	s	s				
	<u>X</u>	f	f	f	<u>f</u>	f	f										
Analog Data	<u>A</u>	n	n	n	l	s	a	a	a	a	a	a	u	u	e	e	
Digital Data	<u>D</u>	a	a	ƀ	d	d	d	d	d	d	d	d	d				

(Underlined characters are those which are always present)

Data Set Header: ddd = Day code (Julian date)
hh = hours
mm = minutes
ss = seconds
ffffff = fixed data (the last four digits are used to represent the year)

Analog Data: nnn = channel number
l = limit alarm;
May take the following values: "> " = upper limit exceeded
"< " = lower limit exceeded
" " = data within limits
s = sign of data
aaaaaa = analog data value, including a decimal point
uu = data units;
May take the following values: " C" = degrees centigrade
" F" = degrees Fahrenheit
" V" = volts
"MV" = millivolts
"**" = error (see error conditions below)
ee = error condition;
May take the following values: "OL" = overload
"BT" = broken thermocouple

Digital Data: aa = data address
ƀ = space
dddddddd = digital data

Table 2
Parameter file (FORØ4.DAT) required to process a portion of the data from the 1981 cruise of the USNS HAYES.

```
SORT PARAMETERS
HAYES 1981 CRUISE
 10
1981   29 THU
1981   29 THU
 40 OUT.TEMP   C   DEG C      2X,F6.1,2X,   1
 27 PWR.SUP.   V   VOLTS      2X,F7.3,1X,   0
 26 POS.PWR.   V   VOLTS      2X,F7.3,1X,   0
 22 OUT.TEMP   V   DEG C      2X,F6.1,2X,   1
 21 REL.HUM.   V  PER CENT    2X,F6.0,2X,   0
 20 PRESSURE  MV   TORR       2X,F6.1,2X,   0
 15 SHP.HEAD  MV   DEGREES    2X,F6.0,2X,   1
 14 SHP.SPD.  MV   KNOTS      2X,F6.0,2X,   1
 13 WND.DIR.   V   DEGREES    2X,F6.0,2X,   1
 12 WND.SPD.   V   KNOTS      2X,F6.0,2X,   1

CALC PARAMETERS


WIND SUBROUTINE
 12 REL.SPD.   KNOTS     2X,F6.0,2X,   0
 13 REL.DIR.   DEGREES   2X,F6.0,2X,   0
 14 SHP.SPD.   KNOTS     2X,F6.0,2X,   0
 15 SHP.HEAD   DEGREES   2X,F6.0,2X,   0
100 ABS.SPD.   KNOTS     2X,F6.0,2X,   1
101 ABS.DIR.   DEGREES   2X,F6.0,2X,   1


MOIST SUBROUTINE
 40 OUT.TEMP   DEG C     2X,F6.1,2X,   0
 20 PRESSURE   TORR      2X,F6.1,2X,   0
 21 REL.HUM.  PER CENT   2X,F6.0,2X,   0
102 H2O VAP     PPMV     2X,F6.0,2X,   1


 22 OUT.TEMP   DEG C     2X,F6.1,2X,   0
 20 PRESSURE    TORR     2X,F6.1,2X,   1
 21 REL.HUM.  PER CENT   2X,F6.0,2X,   1
  0                                    1
```

2) A data set within the specified time window is found.
3) A data set having a date later than the last desired time is encountered.

Cases 1) and 3) cause appropriate error messages to be printed and execution terminates. It is assumed that time monotonically increases between data sets.

In case 2), a data input routine (DATAIN) is executed. Data is read from the current data set and compared with the list of desired inputs, as specified by the original parameter file. If a match is found, the data is stored in an array for further processing; otherwise it is ignored. In either case, data input continues until the required data values have been read. In the event that the start of a new data set is encountered before input of the current set is completed, an abnormal exit from the data input routine occurs and an error message is printed.

Regardless of the mode of exit, the search routine is invoked to locate the next data set. As before, the header is tested and any of the three conditions previously mentioned will halt the search. This time, however, case 2) causes a repeat of the data input routine and case 3) causes a data processing function (MANIP) to be performed. Case 1) still produces an error message, but continues to the data processing step rather than terminating the program.

The data input routine also tests for the following error conditions to the FLAG subroutine:

1) Broken sensor.
2) Overloaded sensor.
3) Value exceeds upper set point.
4) Value below lower set point.

All of these conditions are indicated by flags which are present in the original data from the data logger. These tests may be tailored to other data formats by alteration of the FLAG subroutine.

The data storage array is organized as a two dimensional matrix in which the columns contain data obtained from a particular channel and the rows correspond to different data sets (different times). MANIP accesses a cross reference matrix and determines, for each input channel, the column in which the corresponding data has been stored. This column is processed in accordance with the function specified for that input channel and the resulting value is replaced in the data array.

In general, the function will be different for each type of sensor and will have been chosen so as to convert the data logger output (typically a voltage) into a value of an appropriate physical quantity having the desired units. During this processing step the units of the input quantity (as read from the original data file) are compared with the expected input units (as given in the parameter file) and an error message is produced if a disagreement is found. This message identifies the date, time, and channel for which the error was detected and also shows what units were actually found.

After processing of the data is completed, the output subroutine (DATOUT) is called.

Table 3 shows a sample of the input data obtained from the data logger. The resulting error and warning messages appear in an output file, FORØ6.DAT. To this, DATOUT appends a tabular listing of data from the selected input channels, as shown in Table 4. The year, Julian date, and time are listed in the left columns. For each selected channel, a column of data will be produced having a heading which gives the channel number, sensor identification, and the units. A maximum of twelve channels of data may appear across a line printer page. If more channels were requested, additional pages will be produced, each having the date and time on the left and appropriate column headings across the top.

In addition to selection of the data to be listed, the user formats the output by providing FORTRAN-type format specifications, as desired. A total of ten characters (including spaces) should be specified for each channel which is to be printed.

An additional file, FORØ7.DAT, is also produced by METSRT. This file, which contains all of the processed data plus reference information needed by subsequent programs, was intended to be read only by computer. The format was chosen for compactness and few concessions to human readability have been made. A sample of this output is shown in Table 5.

III. METCLC OPERATION

METCLC, listed in Appendix B, reads both the control file and file FORØ7.DAT and calculates values for the following derived quantities:

1) Absolute wind velocity, expressed as a wind speed (knots) and bearing (degrees referenced to true north).
2) Atmospheric moisture loading, with water vapor concentration in ppm by volume.

For each calculation, the name and channel number for each input and output channel is printed in the summary listing, FORØ6.DAT (Table 6). Any problems encountered (missing channels, for example) are also listed at this point.

The calculated values are then stored in the data array and are available for output in both a tabular form and as an array intended to be read by subsequent programs. As before, any combination of these results may be selected for listing by setting the appropriate print switches. The print formats are specified by the user. An example of this listing is given in Table 7.

To provide versatility and make future alterations simple, calculations have been implemented in separate subroutines.

In general, there may be several alternate sources of data for these subroutines (multiple anemometers or hygrometers may have been used, for

Table 3
An excerpt from the USNS HAYES cruise data file (FORØ5.DAT).
The entire file is over 200 pages long when printed.

```
029:00:00:00
001981
 10 + 1.730 V
 11 + 1.326 V
 12 +0.1114 V
 13 +0.2721 V
 14 + 11.048 V
 15 +173.648 V
 20 + 13.438 V
 21 + 3.326 V
 22 +2.2042 V
 23 +0.0006 V
 24 -0.0004 V
 25 -0.0003 V
 26 +16.026 V
 27 -14.230 V
 28 + 0.176 V
 29 - 0.000 V
 40 +  17.6 C
 41 +  19.6 C

029:01:00:00
001981
10 + 1.730 V
11 + 1.303 V
12 +0.1470 V
13 +0.2677 V
14 + 11.048 V
15 +178.628 V
20 + 19.228 V
21 + 3.633 V
22 +2.3636 V
23 +0.0006 V
24 -0.0004 V
25 -0.0003 V
26 +16.002 V
27 -14.191 V
28 + 0.176 V
29 + 0.000 V
40 +  17.7 C
41 +  19.3 C

029:02:00:00
001981
 10 + 1.713 V
 11 + 1.329 V
 12 +0.0991 V
 13 +0.2630 V
 14 + 11.048 V
 15 +173.378 V
 20 + 19.678 V
 21 + 3.332 V
 22 +2.3366 V
 23 +0.0006 V
 24 -0.0004 V
 25 -0.0004 V
 26 +16.026 V
 27 -14.273 V
 28 + 0.176 V
 29 + 0.000 V
 40 +  17.4 C
 41 +  19.1 C
```

Table 4
The METSRT listing (FORØ6.DAT) generated using the parameters and data file shown in Tables 2 and 3. No warnings or error messages were produced by this data, so that portion of the output is not shown.

[illegible] 1981 CRUISE
DATA FOR THE PERIOD JAN 29, 1981 THROUGH JAN 29, 1981

	CHANNEL 40 OUT.TEMP DEG C	CHANNEL 22 OUT.TEMP DEG C	CHANNEL 15 SHP.HEAD DEGREES	CHANNEL 14 SHP.SPD. KNOTS	CHANNEL 13 WND.DIR DEGREES	CHANNEL 12 WND.SPD. KNOTS
1981						
29						
0	17.6	18.5	179.	1.	272.	10.
100	17.7	18.5	179.	1.	255.	13.
200	17.4	18.2	179.	1.	259.	9.
300	16.9	17.8	179.	1.	271.	8.
400	16.2	17.2	179.	1.	271.	8.
500	16.1	16.9	179.	1.	279.	6.
600	15.5	16.4	179.	1.	233.	7.
700	14.9	15.9	179.	1.	255.	7.
800	14.5	15.5	179.	1.	230.	8.
900	14.8	15.7	179.	1.	275.	11.
1000	14.8	15.7	179.	1.	233.	9.
1100	17.5	18.1	179.	1.	257.	15.
1200	17.1	18.1	179.	0.	243.	3.
1300	16.7	17.7	179.	0.	244.	16.
1400	16.9	17.7	179.	0.	252.	14.
1500	16.8	17.9	179.	0.	252.	14.
1600	15.7	17.7	179.	0.	227.	12.
1700	16.7	17.4	179.	0.	235.	12.
1800	16.7	17.5	179.	0.	231.	15.
1900	16.5	17.6	179.	0.	223.	11.
2000	16.4	17.3	179.	0.	207.	10.
2100	16.4	16.0	179.	0.	139.	8.
2200	16.2	17.5	179.	0.	170.	8.
2300	16.0	16.5	179.	0.	152.	9.

Table 5
Part of the METSRT data dump (FORØ7.DAT) produced from the inputs shown above. This data serves as the input to METCLC.

```
[illegible]
 10  14
1981  29 THU
1981  29 THU
 40 OUT.TEMP   DEG C    2X,F5.1,2X,
 27 EXC.INP.   VOLTS    2X,F7.3,1X,
 26 [illegible]  VOLTS    2X,F7.3,1X,
 22 OUT.TEMP   DEG C    2X,F5.1,2X,
 21 REL.HUM.  PER CENT 2X,F5.0,2X,
 20 PRESSURE    TORR    2X,F6.1,2X,
 15 [illegible] DEGREES 2X,F5.0,2X,
 14 [illegible]  KNOTS    2X,F5.0,2X,
 13 WND.DIR   DEGREES  2X,F5.0,2X,
 12 WND.SPD.   KNOTS    2X,F5.0,2X,
1981  29    0 1981  29  100 1981  29  200 1981  29  300 1981  29  400 1981  29  500
1981  29  600 1981  29  700 1981  29  800 1981  29  900 1981  29 1000 1981  29 1100
1981  29 1200 1981  29 1300 1981  29 1400 1981  29 1500 1981  29 1600 1981  29 1700
1981  29 1800 1981  29 1900 1981  29 2000 1981  29 2100 1981  29 2200 1981  29 2300
CHANNEL NUMBER   40
 .176000E+02  .177000E+02  .174000E+02  .169000E+02  .162000E+02  .161000E+02
 .155000E+02  .149000E+02  .145000E+02  .148000E+02  .145000E+02  .173000E+02
 .171000E+02  .167000E+02  .163000E+02  .158000E+02  .157000E+02  .157000E+02
 .157000E+02  .16?000E+02  .164000E+02  .154000E+02  .152000E+02  .150000E+02
CHANNEL NUMBER   27
-.142500E+02 -.141??0E+02 -.142730E+02 -.142030E+02 -.141570E+02 -.14??80E+02
-.141940E+02 -.145410E+02 -.141930E+02 -.146110E+02 -.143300E+02 -.147110E+02
-.140030E+02 -.139?70E+02 -.142140E+02 -.143590E+02 -.140330E+02 -.140390E+02
-.140670E+02 -.140710E+02 -.141540E+02 -.135550E+02 -.144100E+02 -.140050E+02
CHANNEL NUMBER   26
 .160180E+02  .160020E+02  .160200E+02  .153670E+02  .153390E+02  .160250E+02
 .160260E+02  .160160E+02  .159870E+02  .160100E+02  .153340E+02  .160090E+02
 .160110E+02  .160120E+02  .160050E+02  .159670E+02  .153430E+02  .159900E+02
 .160460E+02  .160180E+02  .150500E+02  .150010E+02  .153250E+02  .160110E+02
CHANNEL NUMBER   22
 .148378E+02  .165296E+02  .181911E+02  .177644E+02  .172356E+02  .163489E+02
 .164??3E+02  .159444E+02  .154639E+02  .156639E+02  .157200E+02  .160533E+02
 .160511E+02  .176?32E+02  .17??77E+02  .173844E+02  .175572E+02  .174467E+02
 .175178E+02  .17?467E+02  .173?22E+02  .173644E+02  .173000E+02  .165200E+02
CHANNEL NUMBER   21
 .768200E+02  .757000E+02  .774000E+02  .762000E+02  .752000E+02  .779600E+02
 .797000E+02  .774000E+02  .807200E+02  .804200E+02  .803200E+02  .742400E+02
 .727400E+02  .734000E+02  .772400E+02  .731000E+02  .733000E+02  .726000E+02
 .774400E+02  .771000E+02  .637600E+02  .680000E+02  .691400E+02  .710000E+02
CHANNEL NUMBER   20
 .759373E+03  .757764E+03  .758942E+03  .759?5?E+03  .759444E+03  .759450E+03
 .759460E+03  .759455E+03  .759470E+03  .759520E+03  .759435E+03  .760435E+03
 .760447E+03  .760447E+03  .7?????E+03  .761786E+03  .761451E+03  .761?76E+03
 .761745E+03  .761736E+03  .7?1?8?E+03  .761304E+03  .761426E+03  .761447E+03
CHANNEL NUMBER   15
 .178640E+03  .178?20E+03  .17?570E+03  .173580E+03  .175380E+03  .178580E+03
 .178870E+03  .17?560E+03  .17?570E+03  .173550E+03  .175540E+03  .179540E+03
 .17?3?0E+03  .178530E+03  .176550E+03  .174560E+03  .175600E+03  .178590E+03
 .178?0?E+03  .179?20E+03  .174600E+03  .176620E+03  .175550E+03  .179570E+03
CHANNEL NUMBER   14
 .110400E+01  .110400E+01  .110400E+01  .110300E+01  .110400E+01  .110300E+01
 .110300E+01  .110300E+01  .110300E+01  .110400E+01  .110300E+01  .110300E+01
 .680000E-01  .680000E-01  .680000E-01  .580000E-01  .5?0000E-01  .680000E-01
 .630000E-01  .930000E-01  .?30000E-01  .540000E-01  .530000E-01  .930000E-01
```

Table 6
The METCLC summary (FORØ6.DAT) which results from the parameter file shown in Table 2 and the data set of Table 5.

WIND SUBROUTINE

ITERATION NUMBER	INPUT: REL WND SPD #	NAME	REL WND DIR #	NAME	ABS SHP SPD #	NAME	ABS SHP DIR #	NAME	OUTPUT: ABS WND SPD #	NAME	ABS WND DIR #	NAME
1	12	REL. SPD	13	REL.DIR.	14	SHP.SPD.	15	SHP.HEAD	100	ABS.SPD.	101	ABS.DIR.

WATER VAPOR SUBROUTINE

ITERATION NUMBER	INPUT: TEMPERATURE #	NAME	AIR PRESSURE #	NAME	REL HUMIDITY #	NAME	OUTPUT: WATER VAPOR #	NAME
1	40	OUT.TEMP	20	PRESSURE	21	REL.HUM.	102	H2O VAP.
2	22	OUT.TEMP	20	PRESSURE	21	REL.HUM.	900	MOIS 2

Table 7
The METCLC data listing (FORØ6.DAT) from the same
run which produced Table 6.

JAIAS 1981 CRUISE
DATA FOR THE PERIOD THU 29, 1981 THROUGH THU 29, 1981

	CHANNEL 21 REL.HUM. PER CENT	CHANNEL 20 PRESSURE TORR	CHANNEL 100 ABS.SPD. KNOTS	CHANNEL 101 ABS.DIR. DEGREES	CHANNEL 102 H2O VAP. PPMV	CHANNEL 900 MOIS 2 PPMV
1981						
29						
0	77.	758.4	10.	84.	13425.	.16E+05
100	77.	758.9	13.	81.	13532.	.16E+05
200	78.	759.0	9.	80.	13442.	.16E+05
300	77.	759.2	8.	82.	14701.	.16E+05
400	78.	759.4	8.	82.	14434.	.15E+05
500	78.	759.4	6.	88.	14263.	.15E+05
600	79.	759.5	7.	93.	13357.	.15E+05
700	80.	759.5	6.	94.	13540.	.14E+05
800	80.	759.5	8.	91.	13231.	.14E+05
900	80.	759.9	11.	89.	13332.	.14E+05
1000	81.	760.4	9.	95.	13540.	.14E+05
1100	74.	760.4	15.	81.	14832.	.15E+05
1200	74.	760.4	3.	60.	14354.	.15E+05
1300	73.	760.4	16.	62.	13339.	.15E+05
1400	73.	761.2	14.	70.	14020.	.15E+05
1500	73.	761.9	14.	51.	13340.	.15E+05
1600	73.	761.5	12.	46.	13312.	.15E+05
1700	73.	761.7	12.	54.	13764.	.14E+05
1800	72.	761.7	15.	49.	13527.	.14E+05
1900	72.	761.7	11.	41.	13386.	.15E+05
2000	69.	761.8	10.	26.	12742.	.14E+05
2100	69.	761.3	8.	7.	12767.	.14E+05
2200	68.	761.4	8.	349.	12518.	.14E+05
2300	71.	761.4	9.	340.	12378.	.13E+05

2 FUNCTION CALLS

example). Accordingly, there is provision for user specification of the inputs for each calculation. In fact, the same calculations can be repeated with different combinations of inputs and the results may be listed for comparison.

Absolute wind velocity is calculated by vector addition of the absolute velocity of the sensor platform (ship) and the relative wind velocity. Subroutine input and output vectors are in the form of a magnitude and a direction. It is assumed that the direction is in degrees from true north (for absolute bearings) or degrees clockwise from the platform velocity vector (for relative bearings) and that speeds are in knots.

Atmospheric moisture loading is calculated[1] as

$$[H_2O] = H \frac{P_s(T_a)}{P_a} 10^4$$

where $[H_2O]$ = water vapor concentration (ppmv); H = relative humidity (%); P_a = ambient pressure (mb); $P_s(T_a)$ = saturation vapor pressure (mb) at ambient temperature T_a (°K).

The saturation vapor pressure may be obtained from

$$P_s(T_a) = P_0 \exp\left[\sum_{n=1}^{4} C_n t^n\right]$$

where P_0 = 1013.25 mb; C_1 = 13.3185; C_2 = -1.9760; C_3 = -0.6445; C_4 = -0.1299 and t is given by

$$t = 1 - \frac{373.15}{T_a}$$

with T_a = ambient temperature (°K).

Inputs to this subroutine are assumed to be in units of torr, per cent, and degrees centigrade for pressure, relative humidity, and temperature, respectively. The output is the water vapor concentration in parts per million by volume.

IV. METPLT OPERATION

METPLT is designed to plot selected subsets of the data contained in FORØ7.DAT according to the specifications given in file FORØ4.DAT. The actual details of generating a plot file are handled by DISSPLA, a package of FORTRAN-callable subroutines provided by Integrated Software Systems

1. G.J. McRae, APCA Journal, 30(4), 394 (1980).

Corporation. Basically, METPLT provides the data needed by the DISSPLA routines. Appendix C contains the source code for METPLT, but not for any of the DISSPLA software.

The parameter file (Table 2) is searched until the "PLOT PARAMETERS" section is found and the title to be used on the output is read. Next, the data file is read and stored in memory. In the event that either of these files is missing, or if the "PLOT" section is not found, an error message will be written into FORØ6.DAT.

The next parameters to be read specify the number of days for which the data is to be plotted on a single page (NDAYS) and the dates of the first and last data which is to be plotted. The set of all plotted data for an NDAYS-long period is referred to as a plot set. Typically, the length of a plot set is seven days so that the plotted output will have one week of data per page. There may be more than NDAYS between the initial and final dates specified, in which case multiple plot sets will be produced. Each plot set may itself involve several pages of output since there are normally only three graphs per page.

We must still specify which data is to be plotted and how it is to be plotted. This is done by providing sets of parameters which define each axis for each plot. Subroutines XAXIN and YAXIN are responsible for reading and storing these parameters.

METPLT first searches for the set of parameters which describes the desired X-axis, then it looks for corresponding Y-axis parameters. For each X-axis, there may be multiple Y-axis specifications so that several different graphs may easily be generated using the same independent variable.

Each axis specification consists of the following nine parameters:

1) Channel number.
2) Channel name.
3) Channel units.
4) Minimum value.
5) Incremental value.
6) Maximum value.
7) Threshold value.
8) Hysteresis parameter.
9) Axis type.

The channel number tells the program which data is to be plotted on the specified axis. In the event that a channel number of zero is given, METPLT will use time, rather than data values, for that axis. In this case, the axis will be labeled with the Julian dates and each day will be labeled at 1200 and 2400 hours. For purposes of axis specification, however, times must be given in minutes.

The channel name and units are used to produce a label for the axis. Minimum, maximum, and incremental values are needed in order to calculate the scale.

Threshold and hysteresis parameters provide increased control over the plotted output. If the value of any coordinate is below the corresponding threshold, plotting of the point will be suppressed. The hysteresis parameters allow points to be suppressed if they lie within a specified "dead band" surrounding the most recently plotted point. Note that, when hysteresis is set to zero on any axis, the dead band area will also be zero and all points will be plotted. In order to prevent this, any of the hysteresis parameters may be set to a negative value and will then be ignored.

The axis type parameter allows the user to select either a linear Y-axis (type = LIN) or a "vector" Y-axis. Only linear X-axes are permitted in the current version of METPLT.

In the vector mode, the two channels representing R- and Θ-components are designated by type = RVEC and type = AVEC, respectively. They are used to generate a vector quantity which is then displayed as an arrow of the appropriate length and direction. The tip of the arrowhead is located at the corresponding X-coordinate for the quantity. For reference, a short arrow (not of unit length) is drawn in the zero degree direction and a scale is provided on the Y-axis. This scale, the axis name, and the axis units are those given for the vector magnitude channel.

The vector plotting mode is useful for representing quantities such as wind direction.

Provisions have been incorporated into METPLT allowing easy program enhancement to include other types of axes, such as logarithmic scales or vectors expressed as X- and Y-components.

Error messages are written into FORØ6.DAT by XAXIN or YAXIN if the axis type is not defined or (in the case of vector axes) if the two types are not self consistent.

At this point, subroutine SETUP determines the first and last dates for the next plot set and searches the ITIME array to locate the corresponding rows in the data matrix. If they cannot be found, an error message is produced and the plot set is skipped. Assuming that the rows have been located, LOAD copies the appropriate data into the X- and Y-arrays needed by the actual curve drawing routines. During the loading process, each datum is checked to see if it is invalid (the character string "----"), if it is below the threshold, or if the hysteresis criterion is not met. In the first two cases, the point will not be plotted and a message to this effect will appear in the plot summary listing (FORØ6.DAT). In the third case, a message will also be printed but the point will not be suppressed unless the hysteresis test fails for all axes.

After all of the graphs on a page have been completed, a page caption is added. The title (specified in the parameter file) is written across the top and a subtitle, giving the initial and final dates of the plot set, appears below the title.

Examples of the printed and plotted output from METPLT are shown in Tables 8 and 9.

Additional pages are produced as necessary in order to graph all of the data in the first plot set. Further plot sets will then be created, each one starting on the Julian date following the end of the previous set.

V. METINP OPERATION

METINP is the only interactive program in the METEOR package. A listing of the FORTRAN program is given in Appendix D and an example of a terminal session is shown in Appendix E. METINP uses this dialog to construct the parameter file, FORØ4.DAT.

Initially, the user is asked to identify the program for which a parameter file is desired. The answer to this question is used to select one of three major subroutines: SORTIN, CALCIN, or PLOTIN. These produce control files for METSRT, METCLC, and METPLT, respectively.

In the case of CALCIN, additional information is requested regarding the specific type of calculation required. Depending on the response, either WNDIN (for wind velocity calculations) or MSTIN (for moisture loading) will be called.

In order to minimize the size of the program, each parameter input routine utilizes the same set of I/O subroutines. Subroutine TTYIN writes a prompter message, reads the user's response, and stores the answer. Subroutine FILE then formats the answer as required for that specific line of the parameter file.

To make these two subroutines more generally useful, they operate on data arrays. For example, the prompter character string and the corresponding input format are contained in arrays PROMPT and FORMIN, respectively, while the user response is stored in INARAY. Each call to TTYIN or to FILE may therefore be tailored to specific requirements by passing the appropriate arrays as arguments in the subroutine call.

After all required information has been obtained, subroutine CHECK writes the parameters to the TTY in exactly the form in which they will appear in the final parameter file. If any changes are required, subroutine EDIT allows the old line to be overwritten by a new one, which is then displayed. When all lines have been verified, the complete set is written onto the disk. If further input is desired, the entire process may be repeated, either for another function for the same program or for a different program.

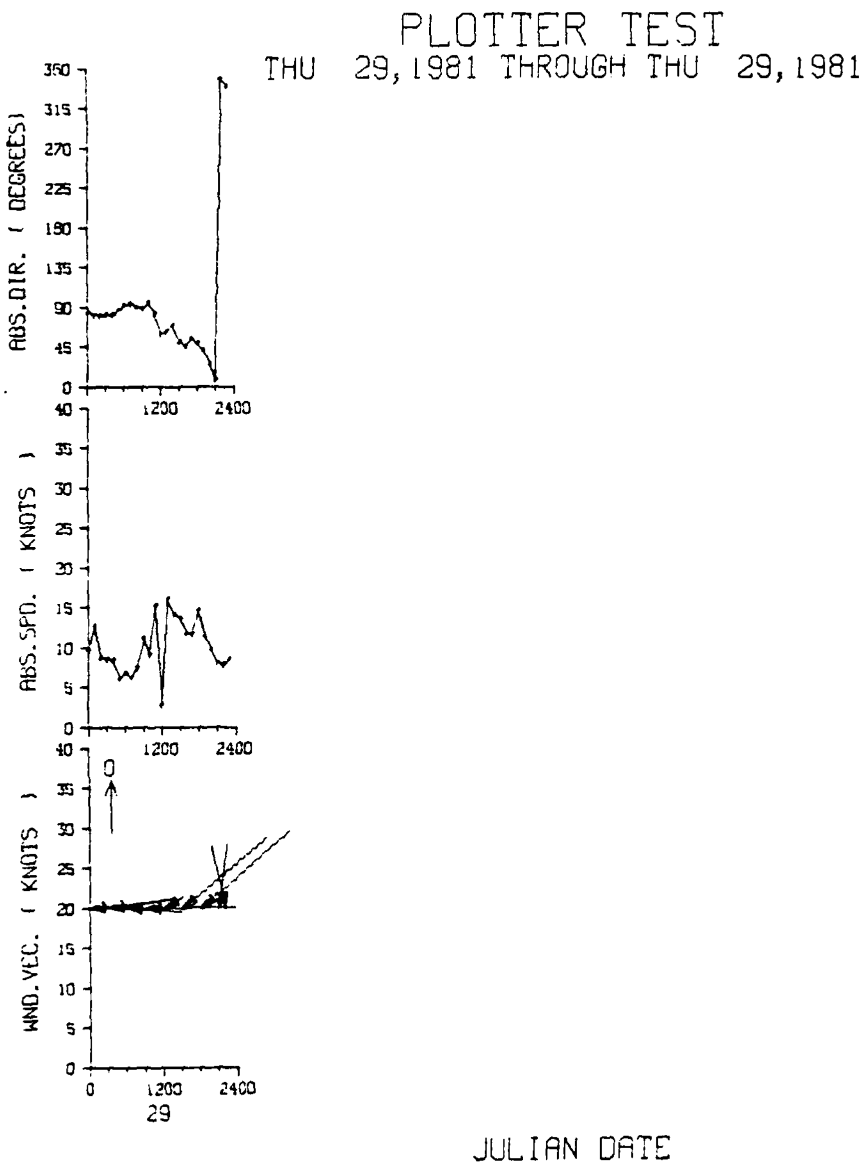
Since METSRT is the first program to be used in data analysis, it is assumed that a new file FORØ4.DAT will be required whenever a METSRT control file is to be created. For this reason, SORTIN causes a new disk file to be opened and any existing file with the name FORØ4.DAT will be destroyed. Some care must therefore be exercised to ensure that the current file (if one exists) has been saved under another name before a new file is opened.

Table 8

The METPLT summary (FORØ6.DAT) resulting from the parameters and data of Tables 2 and 5, respectively.

```
                                      [illegible] 1981 [illegible]
                     DATA FOR THE PERIOD [illegible] 29,1981 [illegible] 29,1981

SUMMARY OF PLOT SET NUMBER   1


PLOT NUMBER   1: WND. VEC.  VS TIME

                 X  CHANNEL =   0; THRESHOLD =     0.; HYSTERESIS =  180.
                 Y  CHANNEL = 100; THRESHOLD =     2.; HYSTERESIS IS UNUSED
                 Y2 CHANNEL = 101, THRESHOLD =     0.; HYSTERESIS =   30.
                 TYPE = VECTOR

1981  29  100: POINT SUPPRESSED --
X  =  .600000E+02 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .613975E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29  200: POINT SUPPRESSED --
X  =  .120000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .790839E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29  400: POINT SUPPRESSED --
X  =  .240000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .615274E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29  500: POINT SUPPRESSED --
X  =  .300000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .678641E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29  700: POINT SUPPRESSED --
X  =  .420000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .930346E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29  800: POINT SUPPRESSED --
X  =  .480000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .903495E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29 1000: POINT SUPPRESSED --
X  =  .600000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .949391E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29 1100: POINT SUPPRESSED --
X  =  .660000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .614630E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29 1300: POINT SUPPRESSED --
X  =  .780000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .604115E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29 1400: POINT SUPPRESSED --
X  =  .840000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .701662E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29 1600: POINT SUPPRESSED --
X  =  .960000E+03 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .455134E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29 1700: POINT SUPPRESSED --
X  =  .102000E+04 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .542041E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29 1900: POINT SUPPRESSED --
X  =  .114000E+04 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .613633E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29 2000: POINT SUPPRESSED --
X  =  .120000E+04 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .657138E+02 IS WITHIN   30. OF PREVIOUS VALUE

1981  29 2300: POINT SUPPRESSED --
X  =  .138000E+04 IS WITHIN  180. OF PREVIOUS VALUE
Y2 =  .300195E+03 IS WITHIN   30. OF PREVIOUS VALUE


PLOT NUMBER   2: AMB. SPD.  VS TIME

                 X  CHANNEL =   0, THRESHOLD =     0.; HYSTERESIS =     0.
                 Y  CHANNEL = 100; THRESHOLD =     [illegible]; HYSTERESIS IS UNUSED
                 TYPE = LINEAR


PLOT NUMBER   3: WND. DIR.  VS TIME

                 X  CHANNEL =   0, THRESHOLD =     0.; HYSTERESIS =     0.
                 Y  CHANNEL = 101; THRESHOLD =     [illegible]; HYSTERESIS IS UNUSED
                 TYPE = LINEAR
```

Table 9
METCLC plotted output corresponding to the above summary.



METCLC and METPLT both require the output of METSRT, so it is assumed, whenever control files for these programs are requested, that the METSRT parameter file already exists. Accordingly, the new parameters are appended to the existing file, which is not lost.

VI. SUMMARY

METEOR provides a coordinated set of programs which can read data tapes, locate specified types of data, test for a wide range of error conditions, calculate values of several derived quantities, and produce both printed and plotted output, all under control of a user-created command file. Existing functions may be selected as required and new functions may be added with relative ease.

Although originally intended to process meteorological data, this software package should be equally applicable to any situation in which large quantities of diverse data are acquired over long time periods.

In many cases,data may be collected on several different data logger systems simultaneously. For these situations, it would be advantageous to be able to merge the resulting data files. Other possible improvements include addition of statistical, curve smoothing, and cross-correlation capabilities in METCLC and provision for better control over plot size and shape in METPLT.

We expect that other users may find it necesssary to revise the METEOR programs to meet their special requirements. It is hoped that the documentation provided will prove to be sufficient for this purpose. Any comments or suggestions regarding alterations to or extension of these programs will be welcomed.

Appendix A
Listing of Program METSRT
(Version 2.0)

PRECEDING PAGE BLANK-NOT FILMED

```
MAIN.    [illegible]                FORTRAN V.[illegible]                              PAGE 1

      PROGRAM NETSET
C
C     NETSET VERSION 2.0   [illegible] 1981
C
C     JOHN HOOVER
C     CODE 41330
C     U.S. NAVAL RESEARCH LABORATORY
C     WASHINGTON, D.C.   20375
C
C     PROGRAM INPUTS DATA [illegible] DATA
C     ACCORDING TO THE DATE [illegible] UNITS, AND
C     [illegible] THE [illegible] ANY OF ALL
C     OF THE DATA MAY ALSO [illegible]
C
C     PROGRAM PARAMETERS:
C
C        NDATA=MAXIMUM POSSIBLE NUMBER OF DATA [illegible] NDATA=300)
C        NCHAN=MAXIMUM POSSIBLE NUMBER OF DATA CHANNELS PER SET (CURRENTLY,
C              NCHAN=25
C
C     IRSTNAME OF THE PARAMETER FILE [illegible] (TITLE=[illegible]
C           FOR THIS PROGRAM)
C
C     INPUT VARIABLES AND FORMATS:
C
C        ITITLE=HEADING [illegible]
C        NCHAN=NUMBER OF DATA CHANNELS [illegible]
C              (FORMAT=1X,[illegible])
C        IYEAR=INITIAL YEAR
C        IDATE=INITIAL JULIAN DATE
C        IDAY=INITIAL DAY OF [illegible]
C        LYEAR,LDATE,LDAY [illegible] LAST YEAR,DATE,DAY
C              (FORMAT=[illegible])
C
C        IREF  =ARRAY WHICH [illegible] DATA [illegible]
C               STORAGE LOCATIONS WITHIN THE DATA ARRAY (COLUMN NUMBER).
C        UNAME =ARRAY OF NAMES OF OUTPUT [illegible]
C        FORM  =ARRAY CONTAINING THE FORMATS FOR [illegible] UNITS
C        IPRINT=ARRAY HOLDING [illegible] ALL DATA CHANNELS
C              (FORMAT=1X,I2,1X,[illegible],1X,[illegible],1X,I7)
C
C
C
C     DATA INPUT :
C
C        DATA  =ARRAY INTO WHICH ALL DATA [illegible] DATA IS ARRANGED
C               IN ROWS [illegible] AND COLUMNS
C               [illegible] INPUT CHANNELS)
C        ITIME =ARRAY HOLDING THE YEAR, DATE, AND [illegible] EACH DATA SET.
C               THE ROWS CORRESPOND TO THE ROWS OF THE DATA MATRIX.
C
C
C     ************************************************************
C                           FILE [illegible]
C                           ---- --------
C
```

```
NETSET [illegible]      FORTRAN V.5A(621) /KI   [illegible]-81      15:17    PAGE 1-1

C     FILE NAME       TYPE             READ              WRITE
C     ---------       ----             ----              -----
C     FOR04.DAT   PARAMETER INPUT     MAIN             *****
C     FOR05.DAT     DATA INPUT      [illegible]        *****
C     FOR06.DAT   FORMATTED LIST      ****          DATAIN,DATOUT,
C                                                    SEARCH,[illegible]
C     FOR07.DAT     DATA OUTPUT       ****             DATOUT
C
C
C     ************************************************************
C                         ERROR MESSAGES
C                         ----- --------
C
C     <NOTE: Y=YEAR,D=DATE,H=HOUR,[illegible] VALUE,I=INTEGER>
C     <       N=CHANNEL NUMBER,[illegible] CHARACTER        >
C
C              MESSAGE              SOURCE           MEANING
C              -------              ------           -------
C     NO PARAMETER FILE FOUND       MAIN       [illegible] WAS MISSING
C                                              [illegible]
C                                              [illegible] COULD BE FOUND
C
C     NO DATA FILE FOUND            MAIN       [illegible] WAS NOT FOUND
C
C     NO DATA SET FOUND             [illegible] NO DATA FOR THE [illegible]
C                                              [illegible] AND DATES COULD BE FOUND
C
C     EOF DURING SEARCH: Y,D,H      SEARCH     EOF WAS ENCOUNTERED WHILE
C                                              SEARCHING FOR NEXT DATA SET.
C                                              [illegible] Y,D, AND H ARE GIVEN
C
C     ERROR DURING SEARCH: Y,D,H    [illegible] [illegible] WHILE AT-
C     -------------------------                [illegible]
C                                              [illegible] Y,D, AND H [illegible]
C                                              [illegible] AND THE [illegible]
C                                              [illegible] (XTOAN)
C
C     SYSTEM RESET AFTER H HOURS ON [illegible] [illegible]
C     DAY D,Y. DATA [illegible]                [illegible]
C                                              [illegible] DUE TO A [illegible]
C                                              [illegible] VALID [illegible]
C                                              [illegible]
C
C     ACQUISITION [illegible] AT H HOURS [illegible] [illegible]
C     ON DAY D,Y.                              [illegible] SYSTEM [illegible]
C                                              [illegible] TO VERIFY
C                                              [illegible] TIME.
C
C     MORE THAN [illegible] DATA [illegible]   [illegible] [illegible]
C     DATA BEYOND H HOURS ON DAY               [illegible]
C     D,Y WILL BE [illegible]
C     LAST DATA [illegible] AT H
C     HOURS ON DAY D,Y
C
C     DUPLICATE DATA FOUND FOR     [illegible] [illegible] DATA [illegible] WITH
C     CHANNEL N AT H HOURS ON                  [illegible] CHANNEL [illegible]
```

```
METSPI  METSPI.FOR     FORTRAN V.5A(621) /KI   [illegible]     15:17    PAGE 1-2

C      DAY D,Y:                                        [illegible]
C      ----------------------------                    [illegible]
C
C      ILLEGAL RECORD AT H HOURS ON       [illegible]  [illegible] AN ALREADY
C      DAY D,Y:                                        DATA RECORD WAS FOUND. [illegible]
C      -----------------------------                   [illegible] RECORD IS [illegible]
C                                                      (4,2044)
C
C      END OF DATA SET [illegible]         [illegible]  [illegible]
C      ON DAY D,Y: N CHANNELS [illegible]              [illegible]
C                                                      [illegible] DATA RECORDS
C
C      INCOMPREHENSIBLE DATA AT            [illegible]  [illegible] CHARACTERS WERE FOUND
C      H HOURS ON DAY D,Y:                             [illegible] DATA [illegible] THE DAT
C      ----------------------------                    [illegible] (14,2044)
C
C      Y,D,H CHANNEL N: [illegible]        [illegible]  "A" DETECTED
C      EXCEEDED
C
C      Y,D,H CHANNEL N: [illegible]        [illegible]  "X" DETECTED
C      EXCEEDED
C
C      Y,D,H CHANNEL N: UNRECOGNIZED       [illegible]  "[illegible]" WAS FOUND IN [illegible]
C      CHARACTER: TEST-__                              OF THE LEGAL CHARACTERS ("B",
C                                                      "O","M").
C
C      Y,D,H CHANNEL N: SENSOR             *[illegible] "B" DETECTED
C      BROKEN
C
C      Y,D,H CHANNEL N: SENSOR             *[illegible] VALID DATA FOUND FOR A SENSOR
C      IS OK                                           PREVIOUSLY FLAGGED AS BROKEN
C
C      Y,D,H CHANNEL N: SENSOR             *[illegible] "O" DETECTED
C      OVERLOAD
C
C      Y,D,H CHANNEL N: SENSOR             *[illegible] VALID DATA FOUND FOR A SENSOR
C      IN RANGE                                        PREVIOUSLY FLAGGED AS [illegible]
C                                                      OVERLOADED
C
C      Y,D,H CHANNEL N: UNRECOGNIZED       [illegible]  "[illegible]" WAS FOUND INSTEAD OF ONE
C      CHARACTER: FLAG=__                              OF THE LEGAL CHARACTERS ("B",
C                                                      "O","M","X")
C
C      ILLEGAL CHANNEL NUMBER N            [illegible]  CHANNEL NO, ENCOUNTERED DURING
C      (ITERATION-I)                                   THE ITH CALL TO "MANIP", IS NOT
C                                                      IN THE RANGE OF VALID CHANNEL
C                                                      NUMBERS
C
C      UNSPECIFIED PROCESS FOR             [illegible]  NO DATA PROCESSING FUNCTION
C      CHANNEL N (ITERATION-I)                         HAS BEEN DEFINED FOR CHANNEL N,
C                                                      ENCOUNTERED DURING THE ITH
C                                                      CALL TO "MANIP"
C
C     *Y,D,H CHANNEL N: UNEXPECTED         *[illegible] THE SPECIFIED DATA WAS IN UNITS
C      UNITS (__) FOUND                                OTHER THAN THOSE EXPECTED
C

METSPI  METSPI.FOR     FORTRAN V.5A(621) /KI   [illegible]     15:17    PAGE 1-3

C      Y,D,H CHANNEL N: CORRECT            *[illegible] CORRECT UNITS WERE FOUND FOR A
C      UNITS (__) FOUND                                CHANNEL PREVIOUSLY FLAGGED AS
C                                                      INCORRECT
C
C      Y,D,H CHANNEL N: HUMIDITY           HU[illegible] HUMIDITY WAS GREATER THAN 100.0
C      OUT OF RANGE (X);                               PERCENT OR LESS THAN 0.0
C      CORRECTED TO X                                  PERCENT
C
C      Y,D,H CHANNEL 25: W.D.              [illegible]  VOLTAGE FOR THE WIND DIRECTION
C      VOLTAGE OUT OF RANGE (X V)                      SENSOR WAS NOT IN THE RANGE
C                                                      [illegible] +- 0.03 VOLTS
C
C      Y,D,H CHANNEL 26: POS.              [illegible]  POSITIVE POWER SUPPLY VOLTAGE
C      PWR. SUPPLY OUT OF RANGE                        WAS NOT IN THE RANGE 15.0
C      (X V)                                           +-0.5 VOLTS
C
C      Y,D,H CHANNEL 27: NEG.              [illegible]  NEGATIVE POWER SUPPLY VOLTAGE
C      PWR. SUPPLY OUT OF RANGE                        WAS NOT IN THE RANGE (-15.0
C      (X V)                                             +-0.5 VOLTS
C
C      *THESE CONDITIONS EXIST FOR ALL SUBSEQUENT DATA FROM THE AFFECTED CHANNEL
C       UNTIL OTHERWISE INDICATED
C
C      ********************************************************************
C
C            !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
C            !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
C            !!                                            !!
C            !!  N     NN    OOOO    TTTTTTTT  EEEEEEEE   !!
C            !!  NN    NN   OOOOOO   TTTTTTTT  EEEEEEEE   !!
C            !!  NNN   NN  OO    OO     TT     EE         !!
C            !!  NNNN  NN  OO    OO     TT     EE         !!
C            !!  NN N  NN  OO    OO     TT     EEEEEE     !!
C            !!  NN  N NN  OO    OO     TT     EEEEEE     !!
C            !!  NN  NNNN  OO    OO     TT     EE         !!
C            !!  NN   NNN  OO    OO     TT     EE         !!
C            !!  NN    NN   OOOOOO      TT     EEEEEEEE   !!
C            !!  NN     N    OOOO       TT     EEEEEEEE   !!
C            !!                                            !!
C            !!  "[illegible]" [illegible]                 !!
C            !!  AND IN ALL [illegible]                     !!
C            !!  [illegible] THAN [illegible]              !!
C            !!  100 DATA [illegible]                      !!
C            !!                                            !!
C            !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
C            !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
C
C      [illegible]
C      METSPI, INIT, [illegible], [illegible], [illegible], [illegible], [illegible], [illegible], [illegible], [illegible],
C      [illegible], ENSUM, [illegible]
C
C      WHEN NEW [illegible] ARE ADDED TO "MANIP", THEY MUST ALSO
C      BE INCLUDED IN THE "[illegible]" [illegible]
C
C      ********************************************************************
C
```

```
      PARAMETER NDAT [illegible]
      DIMENSION [illegible]
      COMMON SPECNT(NCHAN), [illegible]
      COMMON /AFDAT/ DATA( [illegible]
      COMMON /B/ [illegible]
      COMMON /IL/ [illegible]
      COMMON /N/ [illegible]
      COMMON /U/ [illegible]
      DATA [illegible]
      OPEN (UNIT=4, [illegible]
      OPEN (UNIT=4, [illegible]
      OPEN (UNIT=5, [illegible]
      OPEN (UNIT=7, [illegible]
C
C     INITIALIZE A [illegible]
C     CALL INIT
C
C     [illegible]
C
    5 READ (4,10, [illegible]
   10 FORMAT (1A, [illegible]
      IF [illegible]
C
C     INPUT PARAMETERS
C
      READ (4,20) [illegible]
   20 FORMAT [illegible]
      READ (4,25) [illegible]
     1 [illegible]
   25 FORMAT [illegible]
      NACH=0
      NSWICH=0
C
C     INPUT DATA
C
C     LOOK FOR FIRST DATA SET
C
      WRITE (6,30) ITITLE
   30 FORMAT [illegible]
   40 CALL SEARCH(N,690)
      IF (NYEAR-IYEA ) 40,50,70
   50 IF (NDATE-IDATE) 40,60,70
C
C     READ AND STORE DATA
C
   60 CALL DATAIN(N,NSWICH,690)
C
C     LOOK FOR NEXT DATA SET
C
      CALL SEARCH(N,690)
C
C     TEST FOR END OF DATA
C
   70 IF (NYEAR-LYEA ) 60,60,90
   80 IF (NDATE-LDATE) 60,60,90
```

```
   90 IF (NSWICH) 110,110,100
C
C     CONVERT UNITS
C
  100 CALL MANIP
C
C     LIST DATA
C
      CALL DATOUT(NCHAN, [illegible]
C
C     NORMAL EXIT
C
      STOP 'PROGRAM [illegible]'
C
C     ABNORMAL EXITS
C
  110 WRITE (6,120)
  120 FORMAT (A,'NO DATA SET FOUND')
      STOP 'NO DATA SET FOUND'
  130 WRITE (6,140)
  140 FORMAT (A,'NO PARAMETER FILE FOUND')
      STOP 'NO PARAMETER FILE FOUND'
  150 WRITE (6,160)
  160 FORMAT (A,'NO DATA FILE FOUND')
      STOP 'NO DATA FILE FOUND'
      END
```

```
MAIN.   RELST.FOR      FORTRAN V.5A(621) /KI   [illegible]          15:17   PAGE 1

      SUBROUTINE INIT
C
C     SUBROUTINE INITIALIZES [illegible] AND [illegible] TO ZEROS,
C     DATA TO ----, UNIT NAME ARRAY TO BLANKS, AND ASSIGNS DATA
C     ARRAY COLUMN NUMBERS FOR EACH INPUT [illegible]
C
      PARAMETER NDATA = 300,NCHAN = 25
      COMMON IERINT(NCHAN),IERSOL(NCHAN),[illegible]
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible]...SP(NCHAN,4)
      COMMON /U/ UNIT(NDATA,NCHAN,2),UNFM[illegible]
      DATA DASH,BLANK/'----','    '/
      DO 10 J=1,NCHAN
C
C     INITIALIZE ERROR FLAG ARRAYS
C
      IERRNT(J) = 0
      IERSOL(J) = 0
C
C     INITIALIZE COLUMN NUMBERS [illegible]
C
      ISEP(J,4) = J
C
C     INITIALIZE DATA ARRAY
C
      DO 10 I=1,NDATA
      DATA(I,J) = DASH
C
C     INITIALIZE UNIT TO BLANKS
C
      DO 10 K=1,2
   10 UNIT(I,J,K) = BLANK
      RETURN
      END
```

```
SEARCH  RESSST.FOR     FORTRAN V.5A(621) /KI   [illegible]          15:17   PAGE 1

      SUBROUTINE SEARCH(NTIME,*)
C
C     SUBROUTINE SEARCHS FOR AND READS THE [illegible] AND DATE HEADER ON
C     A DATA SET
C
      DIMENSION BADLIN (20)
      COMMON /N/ NYEAR,NDATE,NHOUR,MIN,NEW[illegible]
      DATA Y/'Y   '/
C
C     LOOK FOR DATE AND TIME (START OF DATA SET)
C
   10 READ (5,20,END=50) TEST
   20 FORMAT (A1)
      IF (TEST .NE. Y) GO TO 10
C
C     SAVE PREVIOUS YEAR, DATE, AND TIME
C
      NYEAR = NYEAR
      NDATE = NDATE
      NTIME = NTIME
C
C     READ NEW DATE, TIME, AND YEAR
C
      BACKSPACE 5
      READ (5,30,END=50,ERR=70) NDATE,NHOUR,[illegible]
   30 FORMAT (X,I3,X,I2,X,I2/X,I6)
      NTIME = 100 * NHOUR + MIN
C
C     CHECK FOR SYSTEM RESET
C
      IF (NDATE .EQ. 0) GO TO 100
      IF (IFLAG .EQ. 0) RETURN
      IFLAG = 0
      WRITE (6,40) NTIME,NDATE,NYEAR
   40 FORMAT (X'ACQUISITION RESUMED AT ',I4,' [illegible] DAY ',I3,',',I4)
      RETURN
C
C     IF END OF FILE WAS ENCOUNTERED
C
   50 WRITE (6,60) NYEAR,NDATE,NTIME
   60 FORMAT (1X,'EOF DURING SEARCH: ',I4,[illegible])
      RETURN 1
C
C     IF READ ERROR OCCURRED
C
   70 READ (5,80) BADLIN
   80 FORMAT (20A4)
      WRITE (6,90) NYEAR,NDATE,NTIME,BADLIN
   90 FORMAT (X'ERROR DURING SEARCH: ',I4,[illegible]/A20A4)
      GO TO 10
  100 IF (IFLAG .EQ. 1) GO TO 10
      WRITE (6,110) NTIME,NDATE,NYEAR
  110 FORMAT (X'SYSTEM RESET AFTER ',I4,' [illegible] ',I3,
     1 ',',I4,'. DATA SET(S) SKIPPED.')
      IFLAG = 1
      GO TO 10
      END
```

```
      SUBROUTINE DATAIN(N,[illegible],*)
C
C     SUBROUTINE READS, CHECKS, AND [illegible]
C
      PARAMETER NDATA=100,[illegible]
      DIMENSION BADLIN(20),NAMDAY(7),[illegible]
      COMMON IPRINT(NCHAN),IREAD(NCHAN),[illegible]
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible](NCHAN,4)
      COMMON /II/ IDATE,IYEAR,IDAY,LDATE,[illegible]
      COMMON /N/ NYEAR,NDATE,NHOUR,MIN,[illegible]
      COMMON /L/ UAI [illegible](NDATA,NCHAN,2),[illegible]
      DATA A,I,NAMDAY/'A   ','I   ','SUN ',[illegible],'THU ',
     1 'FRI ','SAT '/
C
C     CLEAR CHANNEL COUNT, INCREMENT ROW [illegible] FOR ROW
C     LIMIT EXCEEDED
C
      NCOUNT = 0
      NROW = NROW + 1
      IF (NROW .LE. NDATA) GO TO 40
C
C     IF LIMIT EXCEEDED, DECREMENT ROW NUMBER [illegible] ERROR MESSAGE
C
      NROW = NROW - 1
      N = NDATA
      WRITE (6,10) N,N,NDATE,[illegible],ITIME(NROW,3),
     1 ITIME(NROW,2),ITIME(NROW,1)
   10 FORMAT (1X,'MORE THAN ',I3,' DATA [illegible]'/1X,'DATA AT ',I4,
     1 ' HOURS ON DAY ',I3,',',I2,' AND [illegible] EXCEEDED'/
     2 1X,'LAST DATA ACCEPTED AT ',I4,' HOURS ON DAY ',I3,',',I2)
C
C     SET LDATE AND LYEAR TO LAST VALUES READ
C
      LDATE = ITIME(NROW,2)
      LYEAR = ITIME(NROW,1)
      N = 1
C
C     FIND INITIAL DAY OF WEEK IN THE [illegible]
C
   20 IF (NAMDAY(N) .EQ. IDAY) GO TO 30
      N = N + 1
      GO TO 20
C
C     FIND NAME OF LAST DAY
C
   30 I = LDATE - IDATE
      N = (I - (I/7)*7) + N
      IF (N .GT. 7) N = N - 7
      IF (N .LT. 1) N = N + 7
      LDAY = NAMDAY(N)
      RETURN 1
C
C     CLEAR "CHANNEL FOUND" FLAG
C
   40 DO 50 I=1,NCHAN
   50 IREAD(I) = 0
C
C     STORE YEAR, DATE, AND TIME
C
      ITIME(NROW,1) = NYEAR
      ITIME(NROW,2) = NDATE
      ITIME(NROW,3) = 100 * NHOUR + MIN
C
C     TEST FOR END OF DATA [illegible]
C
   60 READ (5,70,END=[illegible]) TEST
   70 FORMAT (A1)
C
C     TEST FOR [illegible] DATA
C
      IF (TEST .EQ. A) GO TO 90
C
C     TEST FOR START OF NEXT DATA BLOCK
C
      IF (TEST .EQ. I) GO TO 150
C
C     IF NOT VALID DATA, WRITE ERROR MESSAGE
C
C
      BACKSPACE 5
      READ (5,75) BADLIN
   75 FORMAT (20A4)
      WRITE (6,80) ITIME(NROW,3),NDATE,[illegible]
   80 FORMAT (X,'ILLEGAL RECORD AT ',I4,' HOURS ON DAY ',I3,',',
     1 I4,':'/X,20A4)
      GO TO 60
C
C     IF VALID DATA, REREAD RECORD
C
   90 BACKSPACE 5
      READ (5,100,ERR=190) ICHAN,[illegible]
  100 FORMAT (1X,I3,A1,F7.0,2A2)
C
C     COMPARE WITH ACTIVE CHANNEL LIST
C
      I = 1
  110 IF (ICHAN .EQ. I[illegible](I,1)) GO TO 120
      I = I + 1
      IF (I - NCHAN) 110,110,60
  120 NCOL = IBSP(I,4)
C
C     CHECK FOR DUPLICATE DATA SETS AND [illegible] MESSAGE
C
      IF (IREAD(NCOL) .EQ. 0) GO TO 140
      WRITE (6,130) ICHAN,ITIME(NROW,3),[illegible],A,ICHAN,
     1 TEST,DATUM,UNITS,F1
  130 FORMAT (X,'DUPLICATE DATA FOUND FOR [illegible]',[illegible],' AT ',I4,
     1 ' HOURS ON DAY ',I3,',',I4,':'/
     2 1X,A1,1X,I2,A1,F7.4,2A2)
      GO TO 60
C
C     TEST FOR ERROR CONDITIONS AND SET [illegible]
```

```
DATAIN  [illegible]     FORTRAN [illegible]     [illegible]

  100 CALL FLAG(ICHAN,TEST,[illegible],NROW,NCOL)
C
C     STORE RAW DATA (IF NO ERROR HAS [illegible])
C
      IF (IERROL(NCOL) .EQ. 0 .AND. [illegible])
     * DATA(NROW,NCOL) = DATUM
      UNIT(NROW,NCOL,1) = [illegible]
C
C     SET "CHANNEL FOUND" FLAG, INCREMENT [illegible]
C     "ALL CHANNELS READ"
C
      IREAL(NCOL) = 1
      NCOUNT = NCOUNT + 1
      IF (NCOUNT - NCHAN) 40,170,170
C
C     [illegible] DATA SET WAS [illegible]
C
  170 BACKSPACE [illegible]
      WRITE (6,180) [illegible]
  180 FORMAT (X,'[illegible] DATA SET [illegible]',[illegible],
     1 [illegible],I3,' CHANNELS READ')
C
C     SET "DATA FOUND" [illegible]
C
  190 IF (NCOUNT .GT. 0) [illegible]
  195 RETURN
C
C     [illegible] DURING DATA [illegible]
C
  199 READ(5,200) [illegible]
  200 FORMAT(20A4)
      WRITE(6,210)[illegible]
  210 FORMAT(X,'[illegible] DATA [illegible]',[illegible]
     *I3,[illegible]/1X,20A4)
      GO TO 60
      END
```

```
MAIN.  [illegible]     FORTRAN [illegible]     [illegible]

      SUBROUTINE MANI
C
C     SUBROUTINE CONVERTS DATA TO [illegible]
C
      PARAMETER NDATA=300,NCHAN=[illegible]
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible],IEP(NCHAN,4)
      COMMON /B/ [illegible],NDATA,[illegible]
      COMMON /U/ UNIT(NDATA,NCHAN,2),UNAI[illegible]
      EXTERNAL X1000,X100,PRESST,HUMIDO,[illegible],DIAV1,[illegible],
     1 FREQV,EQUAL,DIV10,X10,DI[illegible]
C
C     CYCLE THROUGH ALL CHANNELS
C
      DO 710 I=1,NCHAN
C
C     LOOK UP CHANNEL AND COLUMN NUMBERS
C
      ICHAN = IEP(I,1)
      NCOL = IEP(I,4)
C
C     SELECT THE PROPER FUNCTION
C
      GO TO (10,10,10,10,10,10,10,10,10,100,[illegible],130,140,150,10,170,
     *140,170,200,210,220,230,240,250,260,270,10,10,300,300,700,10,
     *10,10,10,10,10,10,300,300) ICHAN
C
C     ICHAN OUT OF RANGE; WRITE ERROR MESSAGE
C
      WRITE (6,690) ICHAN,I
  690 FORMAT (1X,'ILLEGAL CHANNEL NUMBER ',[illegible],' (ITERATION = ',I2,')')
      GO TO 710
C
C     NO FUNCTION HAS BEEN DEFINED FOR THIS CHANNEL
C
   10 WRITE (6,700) ICHAN,I
  700 FORMAT (1X,'UNSPECIFIED [illegible] FOR CHANNEL ',I2,' (ITERATION = ',
     1 I2,')')
      GO TO 710
  100 CALL DIDDLE(X10,NROW,NCOL,ICHAN,&710)
  120 CALL DIDDLE(NDAYS,NROW,NCOL,ICHAN,&710)
  130 CALL DIDDLE(DIV2,NROW,NCOL,ICHAN,&710)
  140 CALL DIDDLE(DIV10,NROW,NCOL,ICHAN,&710)
  150 CALL DIDDLE(EQUALC,NROW,NCOL,ICHAN,&710)
  170 CALL DIDDLE(X1000,NROW,NCOL,ICHAN,&710)
  180 CALL DIDDLE(HUMIDO,NROW,NCOL,ICHAN,&710)
  200 CALL DIDDLE(PRESST,NROW,NCOL,ICHAN,&710)
  210 CALL DIDDLE(HUMIDT,NROW,NCOL,ICHAN,&710)
  220 CALL DIDDLE(TEMP1,NROW,NCOL,ICHAN,&710)
  230 CALL DIDDLE(X100,NROW,NCOL,ICHAN,&710)
  240 CALL DIDDLE(DIS1,NROW,NCOL,ICHAN,&710)
  250 CALL DIDDLE(DIAV1,NROW,NCOL,ICHAN,&710)
  260 CALL DIDDLE(POSV,NROW,NCOL,ICHAN,&710)
  270 CALL DIDDLE(FREQV,NROW,NCOL,ICHAN,&710)
  300 CALL DIDDLE(EQUAL,NROW,NCOL,ICHAN,&710)
  710 CONTINUE
  720 RETURN
      END
```

```
      SUBROUTINE DATOUT(NCHPR,NROW,ITITLE)
C
C     SUBROUTINE PRINTS OUT THE RESULTS IN [illegible] FORMATS
C
      PARAMETER NDATA=100,NCHAN=25
      DIMENSION OUT1(40),OUT2(40),BUFFER(1[illegible]
      COMMON IPRINT(NCHAN),IREPRT(NCHAN),[illegible]
      COMMON /REFR/ DATA(NDATA,NCHAN),[illegible],IREF(NCHAN,4)
      COMMON /F/FORM(NCHAN,3)
      COMMON /II/ IDATE,IYEAR,IDAY,[illegible]
      COMMON /U/ UNIT(NDATA,NCHAN,2),UNAME(NCHAN,3)
      DATA C1,C2,FL1,FL2,FB1,X3,A4,DASH/'[illegible]','[illegible] ',' (5X',',I4,',
     *')    ',' ',' A4,','-----'/
C
C     DUMP ALL DATA ONTO DISK (FOR007.DAT) [illegible]
C
      WRITE (7,10) ITITLE,NCHAN,NROW,[illegible],IDATE,IDAY
   10 FORMAT (15A5/2(A13)/(1X,I4,1X,I3,1X,A4))
   15 WRITE (7,20) ((IREF(I,J),J=1,3),(UNAME(I,J),J=1,4),
     1 (FORM(I,J),J=1,3),I=1,NCHAN)
   20 FORMAT (1X,I3,1X,2A4,1X,2A4,1X,3A4)
      WRITE (7,30) ((ITIME(I,J),J=1,3),I=1,NROW)
   30 FORMAT (6(1X,I4,1X,I3,1X,I4))
      DO 45 J=1,NCHAN
      WRITE (7,35) IREF(J,1)
   35 FORMAT (1X,'CHANNEL NUMBER ',I3)
      WRITE (7,40) (DATA(I,J),I=1,NROW)
   40 FORMAT (6(1X,F12.6))
   45 CONTINUE
C
C     LOCATE COLUMNS OF DATA MATRIX WHICH ARE TO BE PRINTED OUT; FIND
C     THE CORRESPONDING CHANNEL NUMBERS, NAMES, AND UNITS.
C     J=OLD COLUMN NUMBER
C     K=NEW COLUMN NUMBER
C
      NPRINT = 0
      J = 1
      K = 1
   60 IF (J - NCHAN) 70,70,85
C
C     PACK INFORMATION OF CHANNELS TO BE PRINTED INTO FIRST N LOCATIONS
C     OF REFERENCE ARRAY
C
   70 IF (IPRINT(J) .EQ. 0) GO TO 80
      UNAME(K,1) = UNAME(J,1)
      UNAME(K,2) = UNAME(J,2)
      IREF(K,1) = IREF(J,1)
      IREF(K,2) = IREF(J,2)
      IREF(K,3) = IREF(J,3)
      IREF(K,4) = IREF(J,4)
      FORM(K,1) = FORM(J,1)
      FORM(K,2) = FORM(J,2)
      FORM(K,3) = FORM(J,3)
      NPRINT = NPRINT + 1
      K = K + 1
   80 J = J + 1
```



```
      GO TO 60
C
C     SET UP TO PRINT N COLUMNS (N<-12)
C
   85 NFIRST = 1
   90 LPRINT = NPRINT
      IF (LPRINT .GT. 12) LPRINT = 12
      NLAST = NFIRST + LPRINT - 1
C
C     WRITE PAGE TITLE
C
      WRITE (6,100) ITITLE,IDAY,IDATE,[illegible],LYEAR
  100 FORMAT (1H1,53X,5A5/30X,'DATA FOR [illegible] ',
     1 A3,1X,I3,',',I6,' THROUGH ',A3,1X,[illegible],',',[illegible]//)
C
C     WRITE CHANNEL NUMBERS, CHANNEL NAMES, AND UNITS
C
      WRITE (6,110) (C1,C2,I=1,LPRINT)
  110 FORMAT (12X,12(2X,2A4))
      WRITE (6,120) (IREF(I,1),I=NFIRST,NLAST)
  120 FORMAT (12X,12(3X,I3,4X)/)
      WRITE (6,110) (IREF(I,2),IREF(I,3),I=NFIRST,NLAST)
      WRITE (6,110) (UNAME(I,1),UNAME(I,2),I=NFIRST,NLAST)
C
C     CONSTRUCT OUTPUT FORMAT ARRAY "OUT1"
C
      OUT1(1) = FL1
      OUT1(2) = FL2
      L = 3
      DO 140 J=NFIRST,NLAST
      DO 140 K=1,3
      OUT1(L) = FORM(J,K)
  140 L = L + 1
      OUT1(L) = FB1
      LMAX = L
C
C     WRITE YEAR, DATE (WHEN DIFFERENT FROM PREVIOUS VALUES),
C     TIME, AND DATA
C
      JYEAR = 0
      JDATE = 0
      DO 190 I=1,NROW
C
C     COPY OUTPUT FORMAT ARRAY INTO "OUT2"
C
      DO 145 L = 1,LMAX
  145 OUT2(L) = OUT1(L)
      K = 1
C
C     LOAD DATA INTO BUFFER AND LOOK FOR [illegible] DATA (----); IF FOUND,
C     ALTER CORRESPONDING LOCATION IN "OUT2"
C
      DO 150 J=NFIRST,NLAST
      BUFFER(K) = DATA(I,IREF(J,4))
      IF (BUFFER(K) .NE. DASH) GO TO 150
      OUT2(3*K) = X3
```

```
          OUT2(3*K + 1) = A4
          OUT2(3*K + 2) = X3
  150     K = K + 1
          IF (ITIME(I,1) .EQ. JYEAR) GO TO 170
          WRITE (6,160) ITIME(I,1)
  160     FORMAT (1X,I4)
          JYEAR = ITIME(I,1)
  170     IF (ITIME(I,2) .EQ. JDATE) GO TO 190
          WRITE (6,180) ITIME(I,2)
  180     FORMAT (5X,I3)
          JDATE = ITIME(I,2)
C
C         WRITE DATA ACCORDING TO "OUT2" AS [illegible] ABOVE
C
  190     WRITE (6,OUT2) ITIME(I,3),(BUFFER(J),J=1,[illegible])
          NPRINT = NPRINT - LPRINT
          NFIRST = NLAST + 1
          IF (NPRINT .GT. 0) GO TO 90
          RETURN
          END
```



```
          SUBROUTINE FLAG(ICHAN,[illegible])
C
C         SUBROUTINE TESTS FOR UPPER OR LOWER [illegible] EXCEEDED AND FOR BROKEN
C         OF OVERLOADED SENSORS. WARNING MESSAGES ARE [illegible] AS NECESSARY.
C
          PARAMETER NDATA=300,NCHAN=25
          COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible],IREF(NCHAN,4)
          COMMON IPRINT(NCHAN),[illegible](NCHAN),[illegible](NCHAN)
          DATA BLANK,PLESS,PMORE,OL,BT/'     ','<    ','>    ','OL   ','BT   '/
          DATA U1,U2,L1,L2/'U    ','[illegible]','L    ','[illegible]'/
          DATA C1,C2,B1,B2/'OVER','LOWE','[illegible]','[illegible]'/
          DATA T1,F1/'TEST','FLAG'/
          DATA NB1,NB2,NO1,NO2/'[illegible]','N    ','[illegible]','[illegible]'/
C
C         TEST FOR UPPER OR LOWER LIMIT EXCEEDED
C
          IF (TEST .EQ. BLANK) GO TO 50
          IF (TEST .EQ. PMORE) GO TO 20
          IF (TEST .NE. PLESS) GO TO 30
C
C         LOWER LIMIT EXCEEDED
C
          WRITE (6,10) (ITIME(NROW,I),I=1,3),[illegible]
   10     FORMAT (1X,I4,1X,I3,1X,I4,' CHANNEL ',[illegible],' : ',A1,A4,
     1    ' LIMIT EXCEEDED.')
          GO TO 50
C
C         UPPER LIMIT EXCEEDED
C
   20     WRITE (6,10) (ITIME(NROW,I),I=1,3),[illegible],U1,U2
          GO TO 50
C
C         ILLEGAL CHARACTER WAS FOUND DURING [illegible]
C
   30     WRITE (6,40) (ITIME(NROW,I),I=1,3),[illegible]
   40     FORMAT (1X,I4,1X,I3,1X,I4,' CHANNEL ',[illegible],' : ',
     1    'UNEXPECTED CHARACTER : ',A4,' = ',[illegible])
C
C         TEST FOR BROKEN OF OVERLOADED SENSORS [illegible] AND MESSAGES PRINTED
C         ACCORDING TO THE FOLLOWING MATRIX:
C
C                                     [illegible] CONDITION
C                  !        OL         !        BT         !       OK
C         ---------!-------------------!-------------------!----------------
C                  !                   !                   !
C                  !                   !  SET BT FLAG      !
C              OL  !    NO CHANGE      !  WRITE "[illegible] RANGE"  !  CLR OL FLAG
C          P C     !                   !  SET OL FLAG      !  WRITE "IN RANGE"
C          R O     !                   !  WRITE "[illegible]"  !
C          E N   --!-------------------!-------------------!----------------
C          V D     !                   !                   !
C          I I     !   CLR BT FLAG     !                   !
C          O T     ! WRITE "NOT BROKEN"!                   !  CLR BT FLAG
C          U I  BT !   SET OL FLAG     !    NO CHANGE      !  WRITE "NOT BROKEN"
C          S O     ! WRITE "OVERLOADED"!                   !
C            N --!-------------------!-------------------!----------------
```

FLAG [illegible] FORTRAN V.3A(1.1) /A1 [illegible] [illegible] PAGE [illegible]

```
C                 !                    !                    !
C              OR !  SET OL FLAG       !  [illegible]       !   NO CHANGE
C                 ! WRITE "OVERLOADED "!  WRITE "[illegible]" !
C     ------------!--------------------!--------------------!--------------------

C
   50 IF (FL .EQ. BLANK) GO TO 80
      IF (FL .EQ. BT) GO TO 70
      IF (FL .NE. OL) GO TO 110
C
C     SENSOR OVERLOADED
C
      IF (IERRBT(NCOL) .EQ. 0) GO TO 100
      IERRBT(NCOL) = 0
      WRITE (6,60) (ITIME(NROW,I),I=1,3),[illegible]
   60 FORMAT (1X,I4,1X,I3,1X,I4,' CHANNEL ',[illegible],' [illegible] ',2A4)
      GO TO 100
C
C     SENSOR BROKEN
C
   70 IF (IERRBT(NCOL) .NE. 0) GO TO 90
      IERRBT(NCOL) = 1
      WRITE (6,60) (ITIME(NROW,I),I=1,3),[illegible]
      GO TO 90
C
C     NO CURRENT ERRORS; CHECK FOR PREVIOUS ERRORS
C
   80 IF (IERRBT(NCOL) .EQ. 0) GO TO 90
      IERRBT(NCOL) = 0
      WRITE (6,60) (ITIME(NROW,I),I=1,3),[illegible]
C
C     TEST FOR PREVIOUS OVERLOAD
C
   90 IF (IERROL(NCOL) .EQ. 0) RETURN
      IERROL(NCOL) = 0
      WRITE (6,60) (ITIME(NROW,I),I=1,3),[illegible]
      RETURN
  100 IF (IERROL(NCOL) .NE. 0) RETURN
      IERROL(NCOL) = 1
      WRITE (6,60) (ITIME(NROW,I),I=1,3),[illegible]
      RETURN
C
C     ILLEGAL CHARACTER WAS FOUND DURING [illegible]
C
  110 WRITE (6,40) (ITIME(NROW,I),I=1,3),[illegible]
      RETURN
      END
```

[illegible] [illegible] FORTRAN V.3A(1.1) /A1 [illegible] 15:17 PAGE 1

```
      SUBROUTINE DIDDLE(PROC,NROW,NCOL,[illegible],*)
C
C     SUBROUTINE DOES DATA CONVERSIONS
C
      PARAMETER NDATA=300,NCHAN=25
      COMMON /ARRAY/ DATA(NDATA,NCHAN),IT[illegible](NDATA,3),IERR(NCHAN,4)
      COMMON /U/ UNIT(NDATA,NCHAN,2),UNAME([illegible],3)
      DATA DASH/'----'/
C
C     CLEAR ERROR FLAG
C
      IFLAG = 0
C
C     OPERATE ON ALL ROWS UNLESS DATA IS BAD (----)
C
      DO 60 I=1,NROW
      IF (DATA(I,NCOL) .EQ. DASH) GO TO 60
      UNITS = UNIT(I,NCOL,1)
C
C     TEST FOR UNEXPECTED UNITS
C
      IF ((UNITS .NE. UNAME(NCOL,3)) .AND. ([illegible] .EQ. 0)) GO TO 10
      IF ((UNITS .EQ. UNAME(NCOL,3)) .AND. ([illegible] .EQ. 1)) GO TO 30
      GO TO 50
C
C     IF UNITS ARE BAD BUT WERE PREVIOUSLY OK, SET FLAG AND WRITE MESSAGE
C
   10 WRITE (6,20) (ITIME(I,J),J=1,3),[illegible]
   20 FORMAT (1X,I4,1X,I3,1X,I4,' CHANNEL ',[illegible],': UNEXPECTED UNITS (',
     1 A2,') FOUND')
      IFLAG = 1
      GO TO 50
C
C     IF UNITS ARE OK BUT WERE PREVIOUSLY BAD, [illegible] FLAG AND WRITE MESSAGE
C
   30 WRITE (6,40) (ITIME(I,J),J=1,3),[illegible]
   40 FORMAT (1X,I4,1X,I3,1X,I4,' CHANNEL ',[illegible],': CORRECT UNITS (',
     1 A2,') FOUND')
      IFLAG = 0
   50 J = I
      DATA(J,NCOL) = PROC([illegible](J,NCOL),J,[illegible])
   60 CONTINUE
      UNIT(1,NCOL,1) = UNAME(NCOL,1)
      UNIT(1,NCOL,2) = UNAME(NCOL,2)
      RETURN 1
      END
```

```
DIV10   METSPT.FOR     FORTRAN V.5A(621) /KI   22-[illegible]-81     15:17   PAGE 1

        FUNCTION DIV10(DATUM,IROW,ICOL)
C
C       FUNCTION CONVERTS MILLIVOLTS TO [illegible] [illegible] A SCALE FACTOR
C       OF ONE TENTH. (NEW CHAN. 14)
C
        DIV10 = DATUM * .1
        RETURN
        END
```

```
X10     METSPT.FOR     FORTRAN V.5A(621) /KI   22-[illegible]-81     15:17   PAGE 1

        FUNCTION X10(DATUM,IROW,ICOL)
C
C       FUNCTION CONVERTS VOLTS TO [illegible] AND [illegible] A SCALE
C       FACTOR OF .01. (NEW CHAN. 10,11)
C
        X10 = DATUM * 10.
        RETURN
        END
```

```
X1000   METSPT.FOR     FORTRAN V.5A(621) /KI   22-[illegible]-81     15:17   PAGE 1

        FUNCTION X1000(DATUM,IROW,ICOL)
C
C       FUNCTION CONVERTS VOLTS TO MILLIVOLTS AND APPLIES A SCALE
C       FACTOR OF ONE. (OLD CHAN. 17,15)
C
        X1000 = DATUM * 1000.
        RETURN
        END
```

```
X100    METSPT.FOR     FORTRAN V.5A(621) /KI   22-[illegible]-81     15:17   PAGE 1

        FUNCTION X100(DATUM,IROW,ICOL)
C
C       FUNCTION CONVERTS VOLTS TO MILLIVOLTS AND APPLIES A SCALE
C       FACTOR OF ONE TENTH. (OLD CHAN. 13)
C
        X100 = DATUM * 100.
        RETURN
        END
```

```
EQUAL   METSPT.FOR     FORTRAN V.5A(621) /KI   22-[illegible]-81     15:17   PAGE 1

        FUNCTION EQUAL(DATUM,IROW,ICOL)
C
C       FUNCTION APPLIES A SCALE FACTOR OF [illegible] (CHAN. 30-32,40,41)
C
        EQUAL = DATUM
        RETURN
        END
```

```
EQUALC  METSPT.FOR     FORTRAN V.5A(621) /KI   22-[illegible]-81     15:17   PAGE 1

        FUNCTION EQUALC(DATUM,IROW,ICOL)
C
C       FUNCTION APPLIES A SCALE FACTOR OF [illegible] AND CORRECTS RANGE
C       0<=EQUALC<=540 TO RANGE 0<=EQUALC<[illegible] (NEW CHAN. 15)
C
        EQUALC = DATUM
   10   IF (EQUALC .LT. 360.) RETURN
        EQUALC = EQUALC - 360.
        GO TO 10
        END
```

```
MMDKTS  METSPT.FOR     FORTRAN V.5A(621) /KI   22-[illegible]-81     15:17   PAGE 1

        FUNCTION MMDKTS(DATUM,IROW,ICOL)
C
C       FUNCTION CONVERTS VOLTS TO MILLIVOLTS AND APPLIES A SCALE
C       FACTOR OF ONE TENTH AND CONVERTS MPH TO [illegible]. (NEW CHAN. 12)
C
        MMDKTS = DATUM * [illegible]
        RETURN
        END
```

```
PRESS1  METSPT.FOR     FORTRAN V.5A(621) /KI   22-[illegible]-81     15:17   PAGE 1

        FUNCTION PRESS1(DATUM,IROW,ICOL)
C
C       FUNCTION TESTS UNITS, CONVERTS VOLTS TO MILLIVOLTS WHERE
C       NECESSARY, AND APPLIES THE EQN. P=(2[illegible]/[illegible])*(MV)+749
C       TORR. (CHAN. 20)
C
        PARAMETER NDATA=300,NCHAN=25
        COMMON /U/ UNIT(NDATA,NCHAN,2),UNAME([illegible])
        DATA V/'V   '/
        IF (UNIT(IROW,ICOL,1) .EQ. V) DATUM = 1000 * DATUM
        PRESS1 = .308 * DATUM + 749.
        RETURN
        END
```

```
HUMIDO  METSFI.FOR      FORTRAN V.5A(621) /KI   [illegible]      15:17   PAGE 1

      FUNCTION HUMIDO(DATUM,IROW,ICOL)
C
C     FUNCTION CONVERTS VOLTS TO MILLIVOLTS, APPLIES A [illegible] FACTOR OF ONE
C     TENTH, AND CALLS "HUMCOR" FOR VALUES [illegible] (OLD CHAN. 14)
C
      HUMIDO = DATUM * 100.
      IF ((HUMIDO .LT. 0.) .OR. (HUMIDO .GT. 100.))
     1 CALL HUMCOR(HUMIDO,IROW,ICOL)
      RETURN
      END
```

```
HUMID1  METSFI.FOR      FORTRAN V.5A(621) /KI   [illegible]      15:17   PAGE 1

      FUNCTION HUMID1(DATUM,IROW,ICOL)
C
C     FUNCTION APPLIES THE EQN. H=20*(V) [illegible] "HUMCOR" FOR
C     VALUES [illegible] >100. (CHAN 21)
C
      PARAMETER NDATA=500,NCHAN=25
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible](NCHAN,4)
      HUMID1 = 20. * DATUM
      IF ((HUMID1 .LT. 0.) .OR. (HUMID1 .GT. 100.))
     1 CALL HUMCOR(HUMID1,[illegible])
      RETURN
      END
```

```
TEMP1   METSFI.FOR      FORTRAN V.5A(621) /KI   [illegible]      15:17   PAGE 1

      FUNCTION TEMP1(DATUM,IROW,ICOL)
C
C     FUNCTION APPLIES EQN. F=40*(V)-50 AND CONVERTS TO DEG. C (CHAN 22)
C
      TEMP1 = 5./9. * (40. * DATUM - 82.)
      RETURN
      END
```

```
DIF1    METSFI.FOR      FORTRAN V.5A(621) /KI   22-[illegible]      15:17   PAGE 1

      FUNCTION DIF1(DATUM,IROW,ICOL)
C
C     FUNCTION APPLIES EQN. D=100*(V);0<[illegible] D=[illegible]100*(V),
C     -1.80<=V<0. (OLD CHAN. 24)
C
      DIF1 = 100. * DATUM
      IF (DATUM .LT. 0) DIF1 = 360. + [illegible]
      RETURN
      END
```

```
DIS2    METSFI.FOR      FORTRAN V.5A(621) /KI   22-[illegible]      15:17   PAGE 1

      FUNCTION DIS2(DATUM,IROW,ICOL)
C
C     FUNCTION CONVERTS VOLTS TO MILLIVOLTS, APPLIES A SCALE
C     FACTOR OF ONE, AND CORRECTS RANGE 0<[illegible] TO RANGE
C     0<=DIS2<360. (NEW CHAN. 13)
C
      DIS2 = 1000. * DATUM
   10 IF (DIS2 .LT. 360.) RETURN
      DIS2 = DIS2 - 360.
      GO TO 10
      END
```

```
DIRV1   METSFI.FOR      FORTRAN V.5A(621) /KI   22-[illegible]      15:17   PAGE 1

      FUNCTION DIRV1(DATUM,IROW,ICOL)
C
C     FUNCTION CHECKS VOLTAGE OF WIND DIRECTION POWER SUPPLY AND
C     ISSUES A WARNING IF IT LIES OUTSIDE THE RANGE 3.60+-0.09 V.
C     (OLD CHAN. 25)
C
      PARAMETER NDATA=500,NCHAN=25
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible](NCHAN,4)
      DIRV1 = DATUM
      IF ((DATUM .LT. 3.57) .OR. (DATUM .GT. [illegible]))
     1 WRITE (6,10) ([illegible](IROW,[illegible]),[illegible]),DATUM
   10 FORMAT (1X,I4,1X,I3,1X,I4,' CHANNEL ',[illegible],
     1 ': W.D. VOLTAGE OUT OF RANGE (',F7.4,' V)')
      RETURN
      END
```

```
POSV    METSFI.FOR      FORTRAN V.5A(621) /KI   22-[illegible]      15:17   PAGE 1

      FUNCTION POSV(DATUM,IROW,ICOL)
C
C     FUNCTION CHECKS VOLTAGE OF THE POSITIVE POWER SUPPLY AND ISSUES
C     A WARNING IF IT LIES OUTSIDE THE RANGE [illegible] (CHAN. [illegible])
C
      PARAMETER NDATA=500,NCHAN=25
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible](NCHAN,4)
      POSV = DATUM
      IF ((DATUM .LT. [illegible]) .OR. (DATUM .GT. [illegible]))
     1 WRITE (6,10) ([illegible](IROW,I),I=1,3),[illegible],DATUM
   10 FORMAT (1X,I4,1X,I3,1X,I4,' CHANNEL ',[illegible],
     1 ': POS. PWR. SUPPLY OUT OF RANGE (',F7.3,' V)')
      RETURN
      END
```

```
ENRGV   METSFI.FOR      FORTRAN V.5A(621) /KI   [illegible]        15:17    PAGE 1

        FUNCTION ENRGV(DATUM,IECH,ICOL)
C
C       FUNCTION CHECKS VOLTAGE OF THE NEGATIVE [illegible] SUPPLY AND ISSUES
C       A WARNING IF IT LIES OUTSIDE THE RANGE [illegible]-2.0 (CHAN. 77)
C
        PARAMETER MDATA=300,MCHAN=[illegible]
        COMMON /ARRAY/ DATA(MDATA,MCHAN),[illegible],ITEP(MCHAN,4)
        ENRGV = DATUM
        IF ((DATUM .GT. -13.0) [illegible] (DATUM [illegible] -17.0))
      1 WRITE (6,10) (ITIME(IECH,[illegible]-1,3),[illegible]),DATUM
     10 FORMAT (1X,I4,[illegible],' CHANNEL ',[illegible],
      1 ': NEG. EXC. SUPPLY OUT OF RANGE (',F7.3,' V)')
        RETURN
        END
```

```
HUMCOR  METSAT.FOR      FORTRAN V.5A(621) /KI   [illegible]        15:17    PAGE 1

        SUBROUTINE HUMCOR(DATUM,IECH,ICOL)
C
C       SUBROUTINE ISSUES A WARNING AND CORRECTS THE DATUM IF HUMIDITY IS FOUND
C       TO BE LESS THAN 0.0 PERCENT OR GREATER THAN 100.0 PERCENT.
C
        PARAMETER MDATA=300,MCHAN=[illegible]
        COMMON /ARRAY/ DATA(MDATA,MCHAN),[illegible],ITEP(MCHAN,4)
        DATA ZERO,HUNDRD/0.0,100.0/
        IF (DATUM .GE. ZERO) GO TO 20
        WRITE (6,10) (ITIME(IECH,I),I=1,3),[illegible],DATUM,ZERO
     10 FORMAT (1X,I4,1X,I3,1X,I4,1X,'CHANNEL ',[illegible],I4,
      1 ': HUMIDITY OUT OF RANGE (',F6.0,'), CORRECTED TO ',F4.0)
        DATUM = ZERO
     20 IF (DATUM .LE. HUNDRD) GO TO 30
        WRITE (6,10) (ITIME(IECH,I),I=1,3),[illegible]),DATUM,HUNDRD
        DATUM = HUNDRD
     30 RETURN
        END
```

Appendix B
Listing of Program METCLC
(Version 1.0)

PRECEDING PAGE BLANK-NOT FILMED

```
      PROGRAM METCLC
C
C     METCLC VERSION 1.0   [illegible]1981
C
C     JOHN HOOVER
C     CODE 4330
C     U.S. NAVAL RESEARCH LABORATORY
C     WASHINGTON, D.C.    20375
C
C
C     PROGRAM READS "METSRT" OUTPUT FILE AND CALCULATES ABSOLUTE
C     WIND VELOCITY, WATER VAPOR LOADING, AND [illegible] CONCENTRATION.
C     THIS DATA IS WRITTEN INTO A FILE [illegible] TO BE READ BY
C     PROGRAM "METPLT". ANY OR ALL OF THE DATA MAY ALSO BE
C     PRINTED.
C
C     PROGRAM PARAMETERS:
C
C        MDATA=MAXIMUM POSSIBLE NUMBER OF DATA [illegible]
C              (CURRENTLY, MDATA=300)
C        MCHAN =MAXIMUM POSSIBLE NUMBER OF DATA CHANNELS PER SET
C              (CURRENTLY, MCHAN=25)
C
C     TEST=NAME OF THE PARAMETER FILE SECTION [illegible] (TEST="CALC"
C         FOR THIS PROGRAM)
C
C     ITITLE=HEADING TO BE USED ON OUTPUT [illegible] (FORMAT=A5)
C
C     WORD=KEYWORD TO SELECT SPECIFIC FUNCTIONS (SEE BELOW FOR LIST OF
C          VALID KEYWORDS) (FORMAT=A5)
C
C     INPUT VARIABLES AND FORMATS:
C
C        NCHAN =NUMBER OF DATA CHANNELS [illegible] FOR CURRENT RUN
C        NROW=NUMBER OF ROWS IN THE DATA ARRAY
C        NXTNUM=NEXT AVAILABLE DEFAULT CHANNEL NUMBER
C        IYEAR=INITIAL YEAR
C        IDATE=INITIAL JULIAN DATE
C        IDAY=INITIAL DAY OF WEEK
C        LYEAR,LDATE,LDAY REFER TO LAST [illegible]
C              (FORMAT=3(X,I3),[illegible])
C
C     DATA  INPUT AND FORMATS:
C
C        IREF  =ARRAY WHICH CROSS REFERENCES CHANNEL NUMBER, DATA TYPE,
C               AND STORAGE LOCATION WITHIN THE DATA ARRAY (COLUMN NUMBER)
C        UNAME =ARRAY OF NAMES OF OUTPUT DATA
C        FORM  =ARRAY CONTAINING THE FORMATS FOR PRINTING RESULTS
C               (FORMAT=X,I3,X,A4,X,2A4,[illegible])
C
C        ITIME =ARRAY HOLDING THE YEAR,DATE, AND TIME FOR EACH
C               DATA SET. THE ROWS CORRESPOND TO THE ROWS OF THE
C               DATA MATRIX.
C               (FORMAT=6(X,I4,X,I3,X,I4))
C
C        DATA  =ARRAY INTO WHICH ALL DATA IS [illegible]. DATA IS ARRANGED
```



```
C               IN ROWS (CORRESPONDING TO [illegible] TIMES) AND COLUMNS
C               (CORRESPONDING TO DIFFERENT [illegible] CHANNELS).
C               (FORMAT=6(X,E12.6))
C
C     KEYWORDS:
C
C     FUNCTIONS WITHIN "METCLC" ARE SELECTED BY USE OF
C     KEYWORDS.  CURRENTLY, THE LEGAL KEYWORDS ARE:
C        "WIND " = SELECT VECTOR ADDITION [illegible] FOR CALCULATION
C                  OF ABSOLUTE WIND VELOCITIES.
C        "MOIST" = SELECT FUNCTION TO CALCULATE ATMOSPHERIC WATER
C                  VAPOR LOADING.
C
C     EACH SUBROUTINE IS RESPONSIBLE FOR [illegible] OF PARAMETERS SPECIFIC TO
C     THAT FUNCTION.  SEE COMMENTS WITHIN THE SUBROUTINES FOR DETAILS.
C
C     ******************************************************************
C
C                          ERROR MESSAGES
C                          ----- --------
C
C              <NOTE:  I = [illegible], N=CHANNEL NUMBER>
C
C              MESSAGE                 [illegible]          MEANING
C              -------                 ------          -------
C     NO PARAMETER FILE FOUND          [illegible]    [illegible] WAS MISSING OR
C                                                  NO SECTION LABELED "CALC
C                                                  PARAMETERS" COULD BE FOUND.
C
C     NO DATA FILE FOUND               [illegible]     FILE FOR07.DAT WAS NOT
C                                                  FOUND
C
C     MAXIMUM NUMBER OF CHANNELS       [illegible]      A ROUTINE WAS CALLED WHICH
C     WOULD BE EXCEEDED BY I                       REQUIRED ADDITIONAL COLUMNS IN
C     CURRENT NUMBER OF CHANNELS= J                THE DATA ARRAY FOR ITS OUTPUT,
C     FURTHER ITERATIONS CANCELLED                 AND THE ARRAY AS DIMENSIONED
C                                                  BY PARAMETER "MCHAN" DID NOT
C                                                  HAVE SUFFICIENT SPACE. PARAM-
C                                                  ETER STATEMENTS IN "METCLC"
C                                                  AND IN "METPLT" SHOULD BE
C                                                  CHANGED BEFORE EXECUTING THE
C                                                  PROGRAM AGAIN.
C
C     REQUESTED CHANNEL (N) DOES NOT   [illegible]     CHANNEL N IS NOT AVAILABLE FOR
C     EXIST. ITERATION ABORTED.                    [illegible] IN OUTPUT
C
C     ******************************************************************
C
C     ALL "PARAMETER" STATEMENTS MUST BE CHANGED IF THERE ARE MORE THAN 25
C     CHANNELS OR 300 DATA SETS!
C
C     PROGRAM UNITS CURRENTLY USING PARAMETER STATEMENTS:
C     METCLC, INIT, DATAIN, WARN, DATOUT, [illegible], [illegible], SETUP, ABSWND,
C     WNDCLC, TSTDAT, MOIST, VALCLC
C
      PARAMETER MDATA=300,MCHAN=25
```

MSTCLC MSTCLC.FOR FORTRAN V.5A(621) /KI 2[illegible] 15:16 PAGE 1-2

```
      DIMENSION ITITLE(5)
      COMMON IPRINT(MCHAN)
      COMMON /ARRAY/ DATA(NDATA,MCHAN),II[illegible],IREF(MCHAN,4)
      COMMON /F/ FORM(MCHAN,3)
      COMMON /IL/ IDATE,IYEAR,IDAY,LDATE,[illegible]
      COMMON /N/ NCHAN,NFCN,NATNIN
      COMMON /U/ UNAME(MCHAN,2)
      COMMON /MAT/ MTCHAN(MCHAN),MICOL(MCHAN)
      EXTERNAL KEYVAL
      DATA CALC/'CALC'/
      OPEN (UNIT=6,DEVICE='DSK',ACCESS='[illegible]',[illegible]'FOR06')
      OPEN (UNIT=4,DEVICE='DSK',ACCESS='[illegible]',[illegible]'FOR04',
     1 ERR=110)
      OPEN (UNIT=7,DEVICE='DSK',ACCESS='[illegible]',[illegible]'FOR07',
     1 ERR=130)
C
C     LOOK FOR "CALC" SECTION OF FOR04.DAT
C
   10 READ (4,20,END=110) WORD
   20 FORMAT (XA5)
      IF (WORD .NE. CALC) GO TO 10
      READ (4,30) ITITLE
   30 FORMAT (X5A5)
      CALL INIT
      KOUNT = 0
      CALL DATAIN(ITITLE)
   40 READ (4,50,END=60) WORD
   50 FORMAT (X,A5)
      WRITE (6,55)
   55 FORMAT (//)
      GO TO (40,80,90,60,40) KEYVAL(WORD)
   60 CALL DATOUT(NCHAN,NFCN,ITITLE)
      WRITE (6,70) KOUNT
   70 FORMAT (/X,I3,' FUNCTION CALLS')
      STOP 'PROGRAM EXECUTES SUCCESSFULLY'
   80 CALL ABSWND(&40)
      KOUNT = KOUNT + 1
      GO TO 40
   90 CALL HOIST(&40)
      KOUNT = KOUNT + 1
      GO TO 40
  110 WRITE (6,120)
  120 FORMAT (X'NO PARAMETER FILE FOUND')
      STOP 'NO PARAMETER FILE FOUND'
  130 WRITE (6,140)
  140 FORMAT (X'NO DATA FILE FOUND')
      STOP 'NO DATA FILE FOUND'
      END
```

MAIN. MSTCLC.FOR FORTRAN V.5A(621) /KI 2[illegible] 15:16 PAGE 1

```
      SUBROUTINE INIT
C
C     SUBROUTINE INITIALIZES DATA [illegible] AND ASSIGNS
C     DATA ARRAY COLUMN NUMBERS FOR EACH [illegible]
C
      PARAMETER NDATA=300,MCHAN=25
      COMMON /ARRAY/ DATA(NDATA,MCHAN),[illegible],3),IREF(MCHAN,4)
      DATA DASH/'----- '/
      DO 10 J=1,MCHAN
C
C     INITIALIZE COLUMN NUMBERS IN REFERENCE ARRAY
C
      IREF(J,4) = J
C
C     INITIALIZE DATA ARRAY
C
      DO 10 I=1,NDATA
   10 DATA(I,J) = DASH
      RETURN
      END
```

```
      SUBROUTINE DATAIN(ITITLE)
C
C     SUBROUTINE READS IN PROGRAM PARAMETERS, [illegible] CORRESPONDING TO EACH
C     DATA SET, AND THE DATA ARRAY, ALL [illegible] UNIT 7.
C
      PARAMETER NDATA=300,NCHAN=25
      DIMENSION ITITLE(5)
      COMMON IPRINT(NCHAN)
      COMMON /ARRAY/ DATA(NDATA,NCHAN),ITIME(NDATA,3),IREP(NCHAN,4)
      COMMON /F/ FORM(NCHAN,3)
      COMMON /IL/ IDATE,IYEAR,IDAY,LDATE,[illegible]
      COMMON /U/ UNAME(NCHAN,2)
      COMMON /N/ NCHAN,NROW,NATNUM
      DATA DASH,IBLANK/'---- ','     '/
C
C     CHECK TO SEE IF A NEW PAGE HEADING [illegible]
C
      ISKIP = 0
      DO 10 I=1,5
   10 IF (ITITLE(I) .NE. IBLANK) ISKIP = 1
      IF (ISKIP .EQ. 1) GO TO 30
C
C     USE OLD PAGE HEADING IF A NEW ONE HAS [illegible]
C
      READ (7,20) ITITLE
   20 FORMAT (X5A5)
      GO TO 35
   30 SKIP RECORD 7
C
C     READ PARAMETERS OUTPUT BY THE [illegible]
C
   35 READ (7,40) NCHAN,NROW,NATNUM,[illegible],IYEAR,
     1 LDATE,IDAY
   40 FORMAT (3(X,I3),2(/X,I4,X,I3,X,A5))
C
C     INITIALIZE DEFAULT CHANNEL NUMBER IF [illegible] WAS SPECIFIED
C
      IF (NATNUM .EQ. 0) NATNUM = 900
C
C     READ IN NAMES OF CHANNELS, UNITS, AND [illegible]
C
      READ (7,50) ((IREP(I,J),J=1,3),(UNAME(I,J),J=1,2),
     1 (FORM(I,J),J=1,3),I=1,NCHAN)
   50 FORMAT (1X,I3,1X,2A4,1X,[illegible],1X,3A4)
C
C     READ IN TIME ARRAY
C
      READ (7,60) ((ITIME(I,J),J=1,3),I=1,[illegible])
   60 FORMAT (6(1X,I4,1X,I3,1X,I4))
C
C     READ IN DATA
C
      DO 80 J=1,NCHAN
      SKIP RECORD 7
      READ (7,70) (DATA(I,J),I=1,NROW)
   70 FORMAT (6(1X,E12.6))
```



```
C
C     LOOK FOR E-FORMAT REPRESENTATION OF '-----' AND REPLACE WITH THE
C     A-FORMAT REPRESENTATION
C
      DO 80 I=1,NROW
   80 IF (DATA(I,J) .EQ. .375691E+16) DATA(I,J) = DASH
      RETURN
      END
```

```
      DIMENSION ITITLE(5)
      COMMON IPRINT(NCHAN)
      COMMON /ARRAY/ DATA(NDATA,NCHAN),I[illegible],I[illegible](NCHAN,4)
      COMMON /F/ FORM(NCHAN,3)
      COMMON /IL/ IDATE,IYEAR,IDAY,LDATE,[illegible]
      COMMON /N/ NCHAN,NECH,NATNUM
      COMMON /U/ UNAME(NCHAN,2)
      COMMON /MAT/ MTCHAN(NCHAN),MICOL(NCHAN)
      EXTERNAL KEYVAL
      DATA CALC/'CALC'/
      OPEN (UNIT=6,DEVICE='DSK',ACCESS='[illegible]',[illegible]'FOR06')
      OPEN (UNIT=4,DEVICE='DSK',ACCESS='[illegible]',[illegible]'FOR04',
     1 ERR=110)
      OPEN (UNIT=7,DEVICE='DSK',ACCESS='[illegible]',[illegible]'FOR07',
     1 ERR=130)
C
C     LOOK FOR "CALC" SECTION OF FOR04.DAT
C
   10 READ (4,20,END=110) WORD
   20 FORMAT (XA5)
      IF (WORD .NE. CALC) GO TO 10
      READ (4,30) ITITLE
   30 FORMAT (X5A5)
      CALL INIT
      KOUNT = 0
      CALL DATAIN(ITITLE)
   40 READ (4,50,END=60) WORD
   50 FORMAT (X,A5)
      WRITE (6,55)
   55 FORMAT (//)
      GO TO (80,80,90,60,40) KEYVAL(WORD)
   60 CALL DATOUT(NCHAN,NECH,ITITLE)
      WRITE (6,70) KOUNT
   70 FORMAT (/X,I3,' FUNCTION CALLS')
      STOP 'PROGRAM EXECUTES SUCCESSFULLY'
   80 CALL ABSWND(640)
      KOUNT = KOUNT + 1
      GO TO 40
   90 CALL MOIST(640)
      KOUNT = KOUNT + 1
      GO TO 40
  110 WRITE (6,120)
  120 FORMAT (X'NO PARAMETER FILE FOUND')
      STOP 'NO PARAMETER FILE FOUND'
  130 WRITE (6,140)
  140 FORMAT (X'NO DATA FILE FOUND')
      STOP 'NO DATA FILE FOUND'
      END
```



```
      SUBROUTINE INIT
C
C     SUBROUTINE INITIALIZES DATA TO '-----' AND ASSIGNS
C     DATA ARRAY COLUMN NUMBERS FOR EACH [illegible]
C
      PARAMETER NDATA=300,NCHAN=25
      COMMON /ARRAY/ DATA(NDATA,NCHAN),I[illegible],IREF(NCHAN,4)
      DATA DASH/'----- '/
      DO 10 J=1,NCHAN
C
C     INITIALIZE COLUMN NUMBERS IN [illegible]
C
      IREF(J,4) = J
C
C     INITIALIZE DATA ARRAY
C
      DO 10 I=1,NDATA
   10 DATA(I,J) = DASH
      RETURN
      END
```

```
WARN     METCLC.FOR     FORTRAN V.5A(621) /KI    20-JUL-81     15:18    PAGE 1

      SUBROUTINE WARN(NEWCOL,*)
C
C     SUBROUTINE WARNS THE USER IF AN ADDITION OF "NEWCOL" COLUMNS WILL
C     EXCEED THE AVAILABLE SPACE IN THE DATA ARRAY.
C
      PARAMETER NDATA=300,MCHAN=25
      COMMON /N/ NCHAN,NREC,NATNUM
      N = NCHAN + NEWCOL - MCHAN
      IF (N .LE. 0) RETURN
      WRITE (6,10) N,NCHAN
   10 FORMAT (1X,'MAXIMUM NUMBER OF CHANNELS WOULD BE EXCEEDED BY ',
     1 I3/10X,'CURRENT NUMBER OF CHANNELS = ',I3/
     2 10X,'FURTHER ITERATIONS CANCELLED')
      RETURN 1
      END
```

```
DATOUT   METCLC.FOR     FORTRAN V.5A(621) /KI    20-JUL-81     15:18    PAGE 1

      SUBROUTINE DATOUT(NCHAN,NROW,ITITLE)
C
C     SUBROUTINE PRINTS OUT THE RESULTS IN SPECIFIED FORMATS
C
      PARAMETER NDATA=300,MCHAN=25
      DIMENSION OUT1(40),OUT2(40),BUFFER(1_),..._(5)
      COMMON IPRINT(MCHAN)
      COMMON /ARRAY/ DATA(NDATA,MCHAN),...(...,3),IREF(MCHAN,4)
      COMMON /F/FORM(MCHAN,3)
      COMMON /II/ IDATE,IYEAR,IDAY,LDATE,LYEAR,LDAY
      COMMON /U/ UNAME(MCHAN,2)
      DATA C1,C2,FL1,FL2,FR1,B3,A4,DASH/'CHAN','NAM ',' (8X',',I4,',
     *')    ',' 3X,',' A4,','----'/
C
C     DUMP ALL DATA ONTO DISK (FOR07.DAT) FOR REFERENCE
C
      CLOSE (UNIT=7,FILE='FOR07',DISPOSE='.....')
      OPEN (UNIT=7,DEVICE='DSK',ACCESS='SEQOUT',FILE='FOR07')
      WRITE (7,10) ITITLE,NCHAN,NROW,NATNUM,...,IDATE,IDAY,
     1 LYEAR,LDATE,LDAY
   10 FORMAT (X5A5/3(X,I5),2(/A,I4,X,I3,X,A3))
   15 WRITE (7,20) ((IREF(I,J),J=1,3),(UNAME(I,J),J=1,2),
     1 (FORM(I,J),J=1,3),I=1,NCHAN)
   20 FORMAT (X,I3,X,2A4,X,2A4,X,3A4)
      WRITE (7,30) ((ITIME(I,J),J=1,3),I=1,NROW)
   30 FORMAT (6(1X,I4,1X,I3,1X,I4))
      DO 45 J=1,NCHAN
      WRITE (7,35) IREF(J,1)
   35 FORMAT (1X,'CHANNEL NUMBER ',I3)
      WRITE (7,40) (DATA(I,J),I=1,NROW)
   40 FORMAT (6(1X,F12.6))
   45 CONTINUE
C
C     LOCATE COLUMNS OF DATA MATRIX WHICH ARE TO BE PRINTED OUT; FIND
C     THE CORRESPONDING CHANNEL NUMBERS, NAMES, AND UNITS.
C     J=OLD COLUMN NUMBER
C     K=NEW COLUMN NUMBER
C
      NPRINT = 0
      J = 1
      K = 1
   60 IF (J - NCHAN) 70,70,85
C
C     PACK INFORMATION RE CHANNELS TO BE PRINTED INTO FIRST N LOCATIONS
C     OF THE REFERENCE ARRAY
C
   70 IF (IPRINT(J) .EQ. 0) GO TO 80
      UNAME(K,1) = UNAME(J,1)
      UNAME(K,2) = UNAME(J,2)
      IREF(K,1) = IREF(J,1)
      IREF(K,2) = IREF(J,2)
      IREF(K,3) = IREF(J,3)
      IREF(K,4) = IREF(J,4)
      FORM(K,1) = FORM(J,1)
      FORM(K,2) = FORM(J,2)
      FORM(K,3) = FORM(J,3)
```

```
DATOUT  NPICLC.FOR     FORTRAN V.5A(621) /KI  22-[illegible]-81     15:18   PAGE 1-1

      NPRINT = NPRINT + 1
      K = K + 1
   80 J = J + 1
      GO TO 60
C
C     SET UP TO PRINT N COLUMNS (N<=12)
C
   85 NFIRST = 1
   90 LPRINT = NPRINT
      IF (LPRINT .GT. 12) LPRINT = 12
      NLAST = NFIRST + LPRINT - 1
C
C     WRITE PAGE TITLE
C
      WRITE (6,100) ITITLE,IDAY,IDATE,IYEAR,[illegible]
  100 FORMAT (1H1,53X,5A5/3HX,'DATA FOR THE [illegible] ',
     1 A3,X,I3,',',I6,' THROUGH ',A3,X,I3,',',[illegible]//)
C
C     WRITE CHANNEL NUMBERS, CHANNEL NAMES, AND [illegible]
C
      WRITE (6,110) (C1,C2,I=1,LPRINT)
  110 FORMAT (12X,12(2X,2A4))
      WRITE (6,120) (IREF(I,1),I=NFIRST,NLAST)
  120 FORMAT (12X,12(3X,I3,4X)/)
      WRITE (6,110) (IREF(I,2),IREF(I,3),[illegible])
      WRITE (6,110) (UNAME(I,1),UNAME(I,2),[illegible])
C
C     CONSTRUCT OUTPUT FORMAT ARRAY "OUT1"
C
      OUT1(1) = FL1
      OUT1(2) = FL2
      L = 3
      DO 140 J=NFIRST,NLAST
      DO 140 K=1,3
      OUT1(L) = FORM(J,K)
  140 L = L + 1
      OUT1(L) = FR1
      LMAX = L
C
C     WRITE YEAR, DATE (WHEN DIFFERENT FROM [illegible] VALUES),
C     TIME, AND DATA
C
      JYEAR = 0
      JDATE = 0
      DO 190 I=1,NROW
C
C     COPY OUTPUT FORMAT ARRAY INTO "OUT2"
C
      DO 145 L = 1,LMAX
  145 OUT2(L) = OUT1(L)
      K = 1
C
C     LOAD DATA INTO BUFFER AND LOOK FOR [illegible] (----); IF FOUND
C     ALTER CORRESPONDING LOCATION IN "OUT2"
C
      DO 150 J=NFIRST,NLAST

DATOUT  NPICLC.FOR     FORTRAN V.5A(621) /KI  22-[illegible]-81     15:18   PAGE 1-2

      BUFFER(K) = DATA(I,IREF(J,4))
      IF (BUFFER(K) .NE. DASH) GO TO 150
      OUT2(3*K) = X3
      OUT2(3*K + 1) = A4
      OUT2(3*K + 2) = X3
  150 K = K + 1
      IF (ITIME(I,1) .EQ. JYEAR) GO TO 170
      WRITE (6,160) ITIME(I,1)
  160 FORMAT (1X,I4)
      JYEAR = ITIME(I,1)
  170 IF (ITIME(I,2) .EQ. JDATE) GO TO 190
      WRITE (6,180) ITIME(I,2)
  180 FORMAT (5X,I3)
      JDATE = ITIME(I,2)
  190 WRITE (6,OUT2) ITIME(I,3),(BUFFER(J),J=1,[illegible])
      NPRINT = NPRINT - LPRINT
      NFIRST = NLAST + 1
      IF (NPRINT .GT. 0) GO TO 90
      RETURN
      END
```

```
KEYVAL  KEYCLC.FOR      FORTRAN V.5A(621) /KI   22-[illegible]-81        15:18    PAGE 1

        FUNCTION KEYVAL(WORD)
C
C       FUNCTION CHECKS THE INPUT WORD AND [illegible] A NUMBER AS FOLLOWS:
C                       WORD            KEYVAL
C                       ----            ------
C                       "    "             5
C                       "PLOT "            4
C                       "HOIST"            3
C                       "WIND "            2
C                       OTHER              1
C
        DATA WIND,HOIST,PLOT,BLANK/'WIND ','HOIST','PLOT ','     '/
        KEYVAL = 1
        IF (WORD .EQ. WIND) KEYVAL = 2
        IF (WORD .EQ. HOIST) KEYVAL = 3
        IF (WORD .EQ. PLOT) KEYVAL = 4
        IF (WORD .EQ. BLANK) KEYVAL = 5
        RETURN
        END
```

```
MAKCOL  [illegible]      FORTRAN V.5A(621) /KI   [illegible]        15:1[illegible]    [illegible]

        SUBROUTINE MAKCOL(J,ID,[illegible],KOUNT)
C
C       SUBROUTINE CREATES NEW DATA COLUMNS [illegible] NAME, UNIT, AND FORMAT
C
        PARAMETER NDATA=300,MCHAN=[illegible]
        DIMENSION NSTRNG(2),UNIT(2)
        COMMON IPRINT(MCHAN)
        COMMON /F/ FORM(MCHAN,3)
        COMMON /U/ UNAME(MCHAN,2)
        COMMON /ARRAY/ DATA(NDATA,MCHAN),[illegible],ITEP(MCHAN,4)
        COMMON /N/ NCHAN,NROW,NXTNUM
        COMMON /MAT/ MTCHAN(MCHAN),MTCOL(MCHAN)
        COMMON /P/ NAME(MCHAN,[illegible]),[illegible](MCHAN,[illegible]),[illegible](MCHAN,[illegible]),[illegible](MCHAN)
        DATA E,X1/'E9.2','X,'/
C
C       INCREMENT NUMBER OF CHANNELS
C
        NCHAN = NCHAN + 1
C
C       IF DESIRED CHANNEL NUMBER IS [illegible], ASSIGN NEXT SEQUENTIAL
C       CHANNEL NUMBER AND INCREMENT NXTNUM
C
        IF (MTCHAN(J) .NE. 0) GO TO 10
        MTCHAN(J) = NXTNUM
        NXTNUM = NXTNUM + 1
C
C       UPDATE ITEP WITH NEW CHANNEL NUMBER [illegible] SET AT
C       INITIALIZATION)
C
   10   MTCOL(J) = NCHAN
        ITEP(NCHAN,1) = MTCHAN(J)
C
C       ASSIGN CHANNEL NAME (DEFAULT NAME IS "[illegible]" WHERE "FUNC" IS THE
C       NAME OF THE REQUESTED FUNCTION AND "N" IS A [illegible] NUMBER)
C
        ENCODE(10,20,NSTRNG) ID,KOUNT
   20   FORMAT (A5,[illegible]X,I2,X)
        ITEP(NCHAN,2) = NSTRNG(1)
        ITEP(NCHAN,3) = NSTRNG(2)
C
C       ASSIGN DEFAULT VALUES FOR OUTPUT UNITS AND FORMAT, SET IPRINT=0
C
        UNAME(NCHAN,1) = UNIT(1)
        UNAME(NCHAN,2) = UNIT(2)
        FORM(NCHAN,1) = X1
        FORM(NCHAN,2) = E
        FORM(NCHAN,3) = X1
        IPRINT(NCHAN) = 0
C
C       OVER RIDE DEFAULT VALUES WITH SPECIFIED VALUES, IF ANY
C
        CALL SETUP(J,MTCOL(J))
        RETURN
        END
```

```
FNDCOL  METCLC.FOR     FORTRAN V.5A(621) /KI  [illegible]     15:16   PAGE 1

      SUBROUTINE FNDCOL(NFIRST,NLAST,*)
C
C     SUBROUTINE LOCATES COLUMNS FOR SPECIFIED CHANNEL NUMBERS AND
C     PRINTS ERROR MESSAGE IF ANY ARE NOT FOUND
C
      PARAMETER NDATA=300,NCHAN=25
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible],IREF(NCHAN,4)
      COMMON /N/ NCHAN,NICO,NAINUM
      COMMON /MAT/ MTCHAN([illegible]),MTCOL([illegible])
      IERROR = 0
      DO 40 J=NFIRST,NLAST
      K = J
      MTCOL(J) = 0
C
C     SEARCH FOR REQUESTED CHANNEL NUMBER IN [illegible] AND SAVE CORRESPONDING
C     COLUMN NUMBER
C
      DO 10 I=1,NCHAN
   10 IF (IREF(I,1) .EQ. MTCHAN(J)) MTCOL(J) = IREF(I,4)
C
C     IF NO CHANNEL WAS FOUND, SET ERROR FLAG AND WRITE ERROR MESSAGE
C
      IF (MTCOL(J) .NE. 0) GO TO 30
      IERROR = 1
      WRITE (6,20) MTCHAN(J)
   20 FORMAT (X,'REQUESTED CHANNEL (',I3,') DOES NOT EXIST.'/
     1 X'ITERATION ABORTED.')
      GO TO 40
C
C     OVERRIDE DEFAULTS (PREVIOUS NAME, UNITS, AND PRINT SWITCH)
C     IF NEW VALUES ARE SPECIFIED
C
   30 CALL SETUP(K,MTCOL(K))
   40 CONTINUE
C
C     IF CHANNEL WAS MISSING, ABORT CURRENT [illegible] AND PROCEED TO THE NEXT
C
      IF (IERROR .EQ. 1) RETURN 1
      RETURN
      END
```

```
SETUP   METCLC.FOR     FORTRAN V.5A(621) /KI  22-[illegible]     15:16   PAGE 1

      SUBROUTINE SETUP(I,ICOL)
C
C      SUBROUTINE SETS VALUES FOR CHANNEL [illegible] UNITS, OUTPUT FORMAT,
C      AND PRINT SWITCH
C
      PARAMETER NDATA=300,NCHAN=25
      COMMON IPRINT(NCHAN)
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible],IREF(NCHAN,4)
      COMMON /U/ UNAME(NCHAN,2)
      COMMON /F/ FORM(NCHAN,3)
      COMMON /N/ NAME(NCHAN,2),OUTU(NCHAN,2),OUTF(NCHAN,3),NPRINT(NCHAN)
      DATA IBLANK,BLANK/'     ','     '/
C
C     REPLACE DEFAULT NAME
C
      IF ((NAME(I,1) .EQ. IBLANK) .AND. (NAME(I,2) .EQ. IBLANK))
     1 GO TO 10
      IREF(ICOL,2) = NAME(I,1)
      IREF(ICOL,3) = NAME(I,2)
C
C     REPLACE DEFAULT OUTPUT UNITS
C
   10 IF ((OUTU(I,1) .EQ. BLANK) .AND. (OUTU(I,2) .EQ. BLANK)) GO TO 20
      UNAME(ICOL,1) = OUTU(I,1)
      UNAME(ICOL,2) = OUTU(I,2)
C
C     REPLACE DEFAULT OUTPUT FORMAT
C
   20 IF ((OUTF(I,1) .EQ. BLANK) .AND. (OUTF(I,2) .EQ. BLANK) .AND.
     1 (OUTF(I,3) .EQ. BLANK)) GO TO 30
      FORM(ICOL,1) = OUTF(I,1)
      FORM(ICOL,2) = OUTF(I,2)
      FORM(ICOL,3) = OUTF(I,3)
C
C     SET PRINT SWITCH
C
   30 IPRINT(ICOL) = NPRINT(I)
      RETURN
      END
```

ABSWND METCLC.FOR   FORTRAN V.[illegible] /AI [illegible]   [illegible] PAGE 1

```
      SUBROUTINE ABSWND(*)
C
C     DO ABSOLUTE WIND VELOCITY CALCULATIONS
C
      PARAMETER NDATA=300,NCHAN=[illegible]
      DIMENSION UNITW1(2),UNITW2(2)
      COMMON /MAT/ MTCHAN(NCHAN),MTCOL(NCHAN)
      COMMON /N/ NAME(NCHAN,[illegible]),[illegible](NCHAN,[illegible]),[illegible](NCHAN,3),PRINT(NCHAN)
      DATA ID,UNITW1,UNITW2,UNITS/'WIND ',' [illegible]','[illegible] ',' DEG',
     1 'REES','_____'/
C
C     WRITE HEADING FOR OUTPUT LISTING
C
      WRITE (6,10) (UNITS,[illegible]=1,10)
   10 FORMAT (60X,'WIND [illegible]'/1[illegible],'____[illegible]____',6A5,
     1 [illegible],2A5,'___[illegible]___',2A5/A'[illegible]',[illegible] AND SPD ',
     2 [illegible] AND DIR [illegible] SHIP SPD [illegible],4X,
     3 'ABS WND SPD',[illegible],'ABS WND DIR'/[illegible],[illegible],4X,'NAME',
     4 [illegible]//)
C
C     VERIFY ROOM TO EXPAND DATA ARRAY [illegible] IF SPACE NOT AVAILABLE
C
      ITER = 0
   20 CALL WAPR(2,670)
      ITER = ITER + 1
C
C     READ FUNCTION PARAMETERS
C
      READ (4,30,END=70) (MTCHAN(I),(NAME([illegible]),[illegible]),(OUTU(I,J),
     1 J=1,2),(OUTP(I,J),J=1,3),[illegible](I),[illegible])
   30 FORMAT (X,I3,X,2A4,X,2A4,X,3A4,X,I1)
C
C     LOCATE INPUT DATA
C
      CALL FNDCOL(1,4,640)
C
C     EXPAND DATA ARRAY; UPDATE [illegible], NCHAN, [illegible]
C
      CALL MAKCOL(5,ID,UNITW1,ITER)
      CALL MAKCOL(6,ID,UNITW2,ITER)
C
C     DO CALCULATIONS
C
      CALL WNDCLC(ITER)
C
C     LOOK FOR NEXT FUNCTION
C
   40 READ (4,50,END=60) WORD
   50 FORMAT (1X,A5)
      IF (KEYVAL(WORD) .EQ. 5) GO TO 40
      BACKSPACE 4
      IF (KEYVAL(WORD) .EQ. 1) GO TO 20
      RETURN
   60 BACKSPACE 4
      RETURN
   70 RETURN 1
      END
```

WNDCLC METCLC.FOR   FORTRAN V.[illegible] /AI [illegible]   [illegible] PAGE [illegible]

```
      SUBROUTINE WNDCLC(KOUNT)
C
C     SUBROUTINE CALCULATES ABSOLUTE WIND [illegible] FROM SHIP VELOCITY
C     AND THE RELATIVE WIND VELOCITY. [illegible] WIND DIRECTION IS
C     REPORTED AS THE DIRECTION FROM WHICH [illegible] COMES, BUT
C     SHIP DIRECTION IS REPORTED AS THE DIRECTION IN WHICH THE SHIP
C     IS HEADING.
C
      PARAMETER NDATA=300,NCHAN=[illegible]
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible],3),IREP(NCHAN,4)
      COMMON /N/ NCHAN,NROW,MAXNUM
      COMMON /MAT/ MTCHAN(NCHAN),MTCOL(NCHAN)
      DATA PI/3.14159/
      DO 40 I=1,NROW
      J = 1
C
C     CHECK FOR INVALID DATA, SET OUTPUTS TO [illegible] IF FOUND
C
      CALL TOTVAL(IERROR,J,4)
      IF (IERROR .EQ. 0) GO TO 20
      SPEED = DASH
      ANGLE = DASH
      GO TO 30
C
C     CALCULATE ABSOLUTE ANGLES IN RADIANS
C
   20 SHPANG = DATA(I,MTCOL(4)) * PI/180.
      RELANG = DATA(I,MTCOL(2)) * PI/180.
      WNDANG = SHPANG + RELANG + PI
C
C     CALCULATE COMPONENTS OF ABSOLUTE WIND [illegible]
C
      YVEL = DATA(I,MTCOL(3))*SIN(SHPANG)+[illegible](1))*SIN(WNDANG)
      XVEL = DATA(I,MTCOL(3))*COS(SHPANG)+[illegible](1))*COS(WNDANG)
C
C     CONVERT TO SPEED AND DIRECTION
C
      SPEED = SQRT(XVEL*XVEL + YVEL*YVEL)
      ANGLE =(ATAN2(YVEL,XVEL) * 180./PI) + 180.
      IF (ANGLE .LT. 0.) ANGLE = ANGLE + 360.
C
C     STORE RESULTS IN DATA ARRAY
C
   30 DATA(I,MTCOL(5)) = SPEED
   40 DATA(I,MTCOL(6)) = ANGLE
C
C     OUTPUT LIST OF CHANNELS USED FOR THIS CALCULATION
C
      WRITE (6,50) KOUNT,(IREP(MTCOL(1),I),I=1,3),
     1 (IREP(MTCOL(2),I),I=1,3),(IREP(MTCOL(3),I),I=1,3),
     2 (IREP(MTCOL(4),I),I=1,3),(IREP(MTCOL(5),I),I=1,3),
     3 (IREP(MTCOL(6),I),I=1,3)
   50 FORMAT (4X,I2,2X,6(4X,I3,X,2A4,4X))
      RETURN
      END
```

```
ISTDAT  [illegible]         FORTRAN V.5A(621) /KI  [illegible]        [illegible]  PAGE 1

      SUBROUTINE ISTDAT(IERROR,I,N)
C
C     SUBROUTINE TESTS FOR BAD DATA (----) AND [illegible] IF
C     IT IS DETECTED
C
      PARAMETER NDATA=300,NCHAN=25
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible](NCHAN,4)
      COMMON /MAT/ MICHAN(NCHAN),MICOL([illegible])
      DATA DASH/'----'/
      IERROR = 0
      DO 10 J=1,N
   10 IF (DATA(I,MICOL(J)) .EQ. DASH) IERROR = 1
      RETURN
      END
```

```
MOIST   METCLC.FOR        FORTRAN V.5A(621) /KI  23-[illegible]        15:16  PAGE 1

      SUBROUTINE MOIST(*)
C
C     DO MOISTURE LOADING CALCULATIONS
C
      PARAMETER NDATA=300,NCHAN=25
      DIMENSION UNITN(2)
      COMMON /MAT/ MICHAN(NCHAN),MICOL(NCHAN)
      COMMON /M/ NAME(NCHAN,2),OUTU(NCHAN,2),[illegible](NCHAN,3),MPRINT(NCHAN)
      DATA ID,BLANK,UNITN,ULINE/'MOIST',' ',' [illegible]','[illegible] ','_____'/
C
C     WRITE HEADING FOR OUTPUT LISTING
C
      WRITE (6,10) (ULINE,I=1,8)
   10 FORMAT(56X,'WATER VAPOR SUBROUTINE'/12X,4A5,'___INPUT___',4A5,
     1 28X,'___OUTPUT___'/X'ITERATION',2X,'[illegible] ',4X,'AIR PRES',
     2 'SURE',8X,'REL HUMIDITY',26X,'WATER [illegible] '/2X,'NUMBER',3(5X,
     3 '#',4X,'NAME',6X),25X'#',4X'NAME'//)
C
C     VERIFY ROOM TO EXPAND DATA ARRAY. [illegible] IF SPACE NOT AVAILABLE
C
      ITER = 0
   20 CALL WARN(1,670)
      ITER = ITER + 1
C
C     READ FUNCTION PARAMETERS
C
      READ (4,30,END=70) (MICHAN(I),(NAME(I,J),J=1,2),(INTU(I,J),J=1,2),
     1 (OUTP(I,J),J=1,3),MPRINT(I),I=1,4)
   30 FORMAT(1X,I3,1X,2A4,1X,2A4,1X,3A4,1X,[illegible])
C
C     LOCATE INPUT DATA
C
      CALL FNDCOL(1,3,640)
C
C     EXPAND DATA ARRAY; UPDATE [illegible], [illegible], [illegible]
C
      CALL NANCOL(4,ID,UNITN,ITER)
C
C     DO CALCULATIONS
C
      CALL VAPCIC(ITER)
C
C     LOOK FOR NEXT FUNCTION
C
   40 READ (4,50,END=60) WORD
   50 FORMAT (1X,A5)
      IF (KEYVAL(WORD) .EQ. 5) GO TO 40
      BACKSPACE 4
      IF (KEYVAL(WORD) .EQ. 1) GO TO 20
      RETURN
   60 BACKSPACE 4
      RETURN
   70 RETURN 1
      END
```

```
VAPCLC  WPTCLC.FOR       FORTRAN V.5A(621) /KI   2[illegible]-81        15:18   PAGE 1

      SUBROUTINE VAPCLC(KOUNT)
C
C     SUBROUTINE CALCULATES WATER VAPOR LOADING IN THE
C     ATMOSPHERE
C
      PARAMETER NDATA=300,NCHAN=25
      COMMON /ARRAY/ DATA(NDATA,NCHAN),ITI[illegible](NDATA,3),IREP(NCHAN,4)
      COMMON /N/ NCHAN,NROW,NA[illegible]NUM
      COMMON /MAT/ NTCHAN(NCHAN),NTCOL(NCHAN)
      DATA DASH/'----'/
      FACTOR = 25./19.
      DO 30 I=1,NROW
      K = I
C
C     CHECK FOR BAD DATA; SET OUTPUT TO "----" [illegible]
C
      CALL TSTDAT(IERROR,K,3)
      IF (IERROR .EQ. 0) GO TO 10
      VAPOR = DASH
      GO TO 20
C
C     CONVERT TEMP TO KELVINS, PRESS TO [illegible]
C
   10 TEMP = DATA(I,NTCOL(1)) + 273.15
      PRESS = DATA(I,NTCOL(2))*FACTOR
C
C     CALCULATE WATER VAPOR CONCENTRATION
C
      T = 1.0 - 373.15/TEMP
      EXION = ((((-0.1299*T)-0.6445)*T-1.[illegible])*T+13.3185)*T
      SAT = 1013.25*EXP(EXION)
      VAPOR = 10000.*(SAT/PRESS)*DATA(I,NTCOL(3))
C
C     STORE RESULTS IN DATA ARRAY
C
   20 DATA(I,NTCOL(4)) = VAPOR
   30 CONTINUE
C
C     OUTPUT LIST OF CHANNELS USED IN THIS [illegible]
C
      WRITE(6,40) KOUNT,(IREP(NTCOL(1),I),I=1,3),(IREP(NTCOL(2),I),
     1 I=1,3), (IREP(NTCOL(3),I),I=1,3),(IREP(NTCOL(4),I),I=1,3)
   40 FORMAT (4X,I2,2X,3(4X,I3,[illegible],2A4,[illegible]),[illegible])
      RETURN
      END
```

Appendix C
Listing of Program METPLT
(Version 3.2)

PRECEDING PAGE BLANK-NOT FILMED

```
      PROGRAM METPLT
C
C     METPLT VERSION 3.0 [illegible] 1981
C
C     JOAN HOOVER
C     CODE 4330
C     U.S. NAVAL RESEARCH LABORATORY
C     WASHINGTON, D.C.   20375
C
C     PROGRAM INPUTS AN ARRAY OF DATA [illegible] BY PROGRAMS
C     METSET AND METCLC FOR DATA DUMPS) [illegible] PRINTS.
C
C     PROGRAM PARAMETERS:
C
C        MDATA=MAXIMUM [illegible] (CURRENTLY, MDATA=300)
C        MCHAN=MAXIMUM [illegible] (CURRENTLY,
C              MCHAN=25)
C
C     INPUTS AND FORMATS:
C
C     CONTROL PARAMETERS [illegible] WHICH PARAMETERS ARE
C     IN FILE FOR04.DAT, WHICH WAS CREATED BY PROGRAM [illegible]. DATA IS READ
C     FROM FILE FOR07.DAT, WHICH IS THE [illegible] PROGRAMS METSET AND
C     METCLC.
C
C     FOR07.DAT:
C
C        NCHAN=NUMBER OF CHANNELS READ ([illegible] IN THE DATA ARRAY)
C        NROW=NUMBER OF ROWS IN THE DATA [illegible]
C        NXTNUM=NEXT NUMBER AVAILABLE [illegible] A NEW CHANNEL
C        IYEAR=INITIAL YEAR IN DATA SET
C        IDATE=INITIAL DATE IN DATA SET
C        IDAY=INITIAL DAY IN DATA SET
C        LYEAR,LDATE,LDAY REFER TO THE LAST [illegible], AND DAY IN THE DATA SET
C              FORMAT = 3(XI3),2(/X[illegible],A[illegible])
C
C        OTITLE=OLD PAGE HEADING TO BE USED [illegible]
C              FORMAT = X24A1
C
C
C
C        IREF=ARRAY WHICH [illegible], NAME, AND
C              COLUMN IN DATA [illegible]
C        UNAME=ARRAY CONTAINING THE NAMES [illegible] UNITS
C        FORM=FORMATS USED FOR [illegible]
C              FORMAT = XI3,[illegible]
C        ITIME=ARRAY CONTAINING THE YEAR, [illegible], AND HOUR FOR EACH SET OF
C              DATA. EACH ROW PERTAINS TO [illegible] PERIOD.
C              FORMAT = 6(XI4,XI3,XI4)
C        DATA=ARRAY OF DATA VALUES. EACH [illegible] TO ONE INPUT CHANNEL,
C              EACH ROW CONTAINS DATA FOR [illegible] PERIOD (WHICH IS DEFINED IN
C              THE CORRESPONDING ROW OF "[illegible]"). DATA IS READ COLUMN BY COLUMN.
C              FORMAT = 6(X,F12.6)
C
C     FOR04.DAT:
C
```



```
C        TITLPL=NEW PAGE HEADING TO BE USED FOR OUTPUT
C              FORMAT = X24A1
C        (NOTE: IF "TITLPL" IS NON-BLANK, IT SUPERSEDES "OTITLE")
C
C        NDAYS=NUMBER OF DAYS TO BE PLOTTED [illegible] GRAPH
C        IPYEAR=INITIAL YEAR TO BE PLOTTED
C        IPDATE=INITIAL DATE TO BE PLOTTED
C        IPDAY=INITIAL DAY OF WEEK TO BE PLOTTED
C        LPYEAR,LPDATE,LPDAY REFER TO THE LAST [illegible],DATE & DAY TO BE PLOTTED
C              FORMAT = XI2,2(/XI4,XI3,XA4)
C
C        NUMX=CHANNEL NUMBER FOR X-AXIS DATA
C        NAMEX=X-AXIS NAME
C        UNITX=X-AXIS UNITS
C        XMIN=MINIMUM VALUES FOR SCALING X-AXIS
C        XINC=DESIRED X-AXIS INCREMENTS
C        XMAX=MAXIMUM VALUES FOR SCALING X-AXIS
C        THLDX=X-THRESHOLD VALUE
C        DPLX=X-AXIS HYSTERESIS PARAMETER [illegible], THIS PARAMETER IS NOT TESTED)
C        TYPEX=X-AXIS TYPE SPECIFICATION
C        THERE ARE SIMILAR PARAMETERS FOR [illegible] Y2-AXES (Y=MAGNITUDE;
C        FOR VECTOR PLOTS, Y2=DIRECTION)
C              FORMAT = XI3,X2A4,X2A4,5([illegible])
C
C     ************************************************************************
C                              ERROR MESSAGES
C                              ----- --------
C
C          <Y=YEAR,D=DATE,W=DAY OF WEEK,[illegible],A=VARIABLE(X,Y,OR Y2)>
C                 <P=PARAMETER,[illegible]DATA VALUE,[illegible]ARBITRARY CHARACTERS>
C
C                MESSAGE                  [illegible]          MEANING
C                -------                  ------          -------
C     NO PARAMETER FILE FOUND             [illegible]     FILE FOR04.DAT WAS NOT FOUND
C                                                         OR NO SECTION LABELED "PLOT"
C                                                         COULD BE FOUND.
C
C     NO DATA FILE FOUND                  [illegible]     FILE FOR07.DAT WAS NOT FOUND
C
C     INITIAL YEAR REQUESTED IS NOT       [illegible]     [illegible] WAS < IYEAR
C     FOUND. FIRST DATE WILL BE W DAY
C     D,Y
C
C     INITIAL DATE REQUESTED IS NOT       [illegible]     IPDATE WAS < IDATE
C     FOUND. FIRST DATE WILL BE W DAY
C     D,Y
C
C     FINAL YEAR REQUESTED IS NOT         [illegible]     [illegible] WAS > LYEAR
C     FOUND. LAST DATE WILL BE W DAY
C     D,Y
C
C     FINAL DATE REQUESTED IS NOT         [illegible]     LPDATE WAS > LDATE
C     FOUND. LAST DATE WILL BE W DAY
C     D,Y
C
C     W IS NOT FOUND IN DAY NAME LIST     [illegible]     W IS NOT ONE OF THE SEVEN
```

```
C                                                       LEGAL NAMES ("SUN ","MON ",
C                                                       "TUE ","WED ","THU","FRI ",
C                                                       "SAT ").
C
C      "_____" IS NOT A VALID X-AXIS        [illegible]    CURRENTLY, ONLY A LINEAR X-
C      TYPE. PLOT IS SKIPPED                            AXIS MAY BE SPECIFIED (TYPE
C                                                       = LIN).
C
C      "_____" IS NOT A VALID Y-AXIS        [illegible]    LEGAL Y-AXIS TYPES ARE:
C      TYPE. PLOT IS SKIPPED                            LINEAR (TYPE = LIN)
C                                                       VECTOR (TYPE - RVEC OR AVEC)
C
C      "_____" AND "_____" ARE NOT          [illegible]    FOR VECTOR AXES, ONE SPECI-
C      COMPATIBLE TYPES. PLOT SKIPPED                   FICATION MUST BE "RVEC" AND
C                                                       THE OTHER "AVEC" (MAGNITUDE
C                                                       AND DIRECTION, RESPECTIVELY)
C
C      INITIAL JULIAN DATE (N,D) IS         [illegible]    DATE ON WHICH PLOT SET IS TO
C      NOT FOUND                                        BEGIN WAS NOT FOUND IN ARRAY
C      PLOT SET IS SKIPPED.
C                                                       "ITIME"
C
C      FINAL JULIAN DATE (N,D) IS           [illegible]    DATE ON WHICH PLOT SET IS TO
C      NOT FOUND                                        END WAS NOT FOUND IN "ITIME"
C      PLOT SET IS SKIPPED.
C
C      NUM WAS NOT FOUND IN "LABE"--        [illegible]    THE CHANNEL SPECIFIED FOR
C      PLOT NUMBER N IS SKIPPED                         THE AXIS WAS NOT FOUND.
C                                                       "NUM" (MAY BE "NUMX","NUMY",
C                                                       OR "NUMY2") TELLS WHICH AXIS
C
C      Y,D,T: DATA IS INVALID               [illegible]    A DATA VALUE OF "____" WAS
C                                                       FOUND. POINT NOT PLOTTED.
C
C      Y,D,T: A=X IS < THRESHOLD            [illegible]    A DATA VALUE BELOW THRESHOLD
C                                                       WAS FOUND. POINT NOT PLOTTED
C
C      Y,D,T: POINT SUPPRESSED --           [illegible]    DATA VALUE DOES NOT MEET
C      A=X IS WITHIN P OF PREVIOUS                      THE HYSTERESIS CRITERION.
C      VALUE                                            POINT WILL STILL BE PLOTTED
C                                                       UNLESS ALL TESTS FAIL.
C
C      NOTE: OTHER ERROR MESSAGES MAY BE [illegible] THE "DISSPLA" PACKAGE.
C
C      *****************************************************************
C
       PARAMETER NDATA=300,MCHAN=25
       DIMENSION ITITLE(5),TITLPL(25)
       REAL NAMEX,NAMEY,NAMEY2
       COMMON NUMX(MCHAN),NAMEX(2,MCHAN),[illegible],XMIN(MCHAN),
      1 XINC(MCHAN),XMAX(MCHAN),THLDX(MCHAN),[illegible],TYPEX(MCHAN),
      2 NUMY(MCHAN),NAMEY(2,MCHAN),UNITY(2,[illegible]),[illegible](MCHAN),
      3 YINC(MCHAN),YMAX(MCHAN),THLDY(MCHAN),[illegible],TYPEY(MCHAN),
      4 NUMY2(MCHAN),NAMEY2(2,MCHAN),UNIT[illegible],YMIN2(MCHAN),
      5 YINC2(MCHAN),YMAX2(MCHAN),THLDY2([illegible]),[illegible](MCHAN),
      6 TYPEY2(MCHAN)
```



```
       COMMON /ARRAY/ DATA(NDATA,MCHAN),ITIME([illegible],3),IPEF(MCHAN,4)
       COMMON /DAY/ NAMDAY(7),NUMDAY,DAYSIZ
       COMMON /REAL/ FORM(MCHAN,3),UNAME([illegible])
       COMMON /IL/ IDATE,IYEAR,IDAY,LDATE,[illegible],[illegible]
       COMMON /N/ NCHAN,NROW,NXTNUM,NDAYS
       COMMON /P/ IPDATE,IPYEAR,IPDAY,LPDATE,[illegible],LPDAY
       COMMON /JK/ JPDATE,JPYEAR,JPDAY,KPDA[illegible],[illegible],KPDAY
       COMMON /INTPLT/ IPLOT,LPLOT,NPLOT
       COMMON /RC/ IROW,KROW,LROW,ICOLX,[illegible],[illegible]
       COMMON /ALPHA/ XSTRNG(9),YSTRNG(9)
       COMMON /PCINT/ XARRAY(NDATA),YMAG([illegible]),[illegible](NDATA)
       COMMON /X/ X00,X12,X24,ADATE
       COMMON /SIZE/ XSIZE,YSIZE
       DATA BLANK,DOLLAR/'    ','$'/
       DATA NAMDAY /'SUN','MON','TUE','WED','THU','FRI','SAT'/
       DATA DAYSIZ,NSIZE,YSIZE,[illegible],DASH,[illegible]
      1 / 1.375,9.625,3.0,0.2,'----','PLOT'/
       OPEN (UNIT=6,DEVICE='DSK',ACCESS='[illegible]',[illegible]='FOR06')
       OPEN (UNIT=4,DEVICE='DSK',ACCESS='[illegible]',[illegible]='FOR04',FPP=330)
       OPEN (UNIT=7,DEVICE='DSK',ACCESS='[illegible]',[illegible]='FOR07',FPP=350)
C
C      LOOK FOR "PLOT" SECTION OF FOR04.DA.
C
    10 READ (4,20,END=330) WORD
    20 FORMAT (XA5)
       IF (WORD .NE. PLOT) GO TO 10
C
C      INITIALIZE ARRAY "TITLPL"
C
       DO 25 I=1,25
       TITLPL(I) = BLANK
    25 CONTINUE
       READ (4,30) (TITLPL(I),I=1,24)
    30 FORMAT (X24A1)
C
C      READ FOR07.DAT DATA
C
       CALL DATAIN(TITLPL)
C
C      ENCODE TITLE STRING
C
       N = 25
    35 N = N - 1
       IF (N .EQ. 0) GO TO 40
       IF (TITLPL(N) .EQ. BLANK) GO TO 35
    40 ENCODE (N+1,45,ITITLE) (TITLPL(I),I=1,[illegible])
    45 FORMAT (25A1)
C
C      READ DATES TO BE PLOTTED AND NUMBER OF [illegible]
C
       READ (4,50) NDAYS,IPYEAR,IPDATE,IPDAY,[illegible],[illegible],LPDAY
    50 FORMAT(XI2/XI4,XI3,XA4/XI4,XI3,XA4)
C
C      CONFIRM PRESENCE OF DATA FOR THE [illegible]
C
C      CHECK INITIAL YEAR
```

```
C
      IF (IPYEAR .GE. IYEAR) GO TO 60
      WRITE (6,55) IDAY,IDATE,IYEAR
   55 FORMAT (X'INITIAL YEAR REQUESTED IS NOT FOUND. FIRST DATE WILL ',
     1 'BE ',A4,' DAY ',I3,',',I4)
      IPYEAR = IYEAR
      IPDATE = IDATE
      IPDAY = IDAY
      GO TO 80
C
C     CHECK INITIAL DATE
C
   60 IF (IPDATE .GE. IDATE) GO TO 80
      WRITE (6,70) IDAY,IDATE,IPYEAR
   70 FORMAT (X'INITIAL DATE REQUESTED IS NOT FOUND. FIRST DATE WILL ',
     1 'BE ',A4,' DAY ',I3,',',I4)
      IPDATE = IDATE
      IPDAY = IDAY
C
C     CHECK FINAL YEAR
C
   80 IF (LPYEAR .LE. LYEAR) GO TO 100
      WRITE (6,90) LDAY,LDATE,LYEAR
   90 FORMAT (X'FINAL YEAR REQUESTED IS NOT FOUND. LAST DATE WILL BE ',
     1 A4,' DAY ',I3,',',I4)
      LPYEAR = LYEAR
      LPDATE = LDATE
      LPDAY = LDAY
      GO TO 120
C
C     CHECK FINAL DATE
C
  100 IF (LPDATE .LE. LDATE) GO TO 120
      WRITE (6,110) LDAY,LDATE,LPYEAR
  110 FORMAT (X'FINAL DATE REQUESTED IS NOT FOUND. LAST DATE WILL BE ',
     1 A4,' DAY ',I3,',',I4)
      LPDATE = LDATE
      LPDAY = LDAY
C
C     INITIALIZE SYSTEM
C
  120 CALL INIT (&370)
C
C     LOOK FOR "XAXIS" AND READ PARAMETERS
C
  130 READ (4,140,END=180) WORD
  140 FORMAT (XA5)
      IF (KEYVAL(WORD) .NE. 2) GO TO 130
  150 CALL XAXIN(&130)
C
C     LOOK FOR "YAXIS" AND READ PARAMETERS
C
  160 READ (4,140,END=180) WORD
      GO TO (160,150,170,160) KEYVAL(WORD)
  170 CALL YAXIN (&160)
  180 NPLOT = NPLOT - 1
```



```
      IF (NPLOT .LE. 0) GO TO 370
C
C     SET LENGTH OF X-AXIS
C
      XSIZE = NDAYS * DAYSIZ
      IF (XSIZE .GT. HSIZE) XSIZE = HSIZE
C
C     INITIALIZE PLOTTER AND PLOTTER SOFTWARE
C
      CALL CALCMP
      CALL INCH30
      CALL PAGE(11.0,11.0)
      CALL MARKER(5)
      CALL SCLPIC(0.4)
      CALL INTAXS
      CALL YAXANG(0.0)
      CALL NOCHEK
      CALL XTICKS(4)
      CALL HEIGHT(0.10)
      X00 = -XINT(0)/2.0
      X12 = (DAYSIZ - XINT(1200))/2.0
      X24 = XSIZE - XINT(2400)/2.0
      CALL RESET('HEIGHT')
      XDATE - X12 + 1.0
C
C     SET UP LIMITING DATES AND ROW NUMBERS FOR THE CURRENT PLOT SET.
C
      NSET = 0
  190 NSET = NSET + 1
      CALL SETUP(NSET,TITLPL,&320)
      KPLOT = NPLOT
C
C     DO PLOTS IN SETS OF THREE
C
      IPLOT = 0
  200 LPLOT = NPLOT
      IF (LPLOT .GT. 3) LPLOT = 3
      CALL PHYSOR(1.0,1.0)
      DO 310 I=1,LPLOT
      L = I
      IPLOT = IPLOT + 1
      CALL LOAD (IPLOT,&310)
C
C     SET UP X-AXIS TITLE
C
      IF (ICOLX) 230,210,230
  210 ENCODE (19,220,XSTRNG)
  220 FORMAT (19X)
      GO TO 250
  230 ENCODE (19,240,XSTRNG) NAMEX(1,IPLOT),NAMEX(2,IPLOT),
     1 UNITX(1,IPLOT),UNITX(2,IPLOT)
  240 FORMAT (2A4,' (',2A4,')')
C
C     SET UP Y-AXIS TITLE (USE NAME AND UNITS ASSOCIATED WITH "YMAG").
C
  250 ENCODE (19,240,YSTRNG) NAMEY(1,IPLOT),NAMEY(2,IPLOT),
```

```
METPLT  METPLT.FOR      FORTRAN V.5A(621) /KI   [illegible]        8:22    PAGE 1-6

      1 UNITY(1,IPLOT),UNITY(2,IPLOT)
C
C     SPECIFY SUBPLOT SIZE AND GENERATE [illegible]
C
      CALL TITLE(0,0,XSTRNG,19,[illegible],19,[illegible])
      IF (ICOLX .EQ. 0) CALL XNONUM
      GO TO (310,260,310,310,310,270,310) [illegible](IPLOT))
  260 CALL STRPLT (IPLOT,6300)
  270 CALL VECPLT (IPLOT,6300)
  300 IF (ICOLX .EQ. 0) CALL [illegible]
      CALL RESET ('XNONUM')
      CALL ENDGR (IPLOT)
      CALL OREL(0.0,YSIZE + [illegible])
  310 CONTINUE
      CALL PHYSOR(0.,0.)
      CALL AREA2D(11.,11.)
      CALL TEXT(ITITLE)
      IF (ICOLX .EQ. 0) CALL [illegible]
      CALL ENDPL(0)
      KPLOT = KPLOT - IPLOT
      IF (KPLOT .GT. 0) GO TO 200
  320 IF (KPDATE .LT. LPDATE) GO TO [illegible]
      CALL DONEPL
      STOP 'PROGRAM EXECUTES SUCCESSFULLY'
  330 WRITE(6,340)
  340 FORMAT (X'NO PARAMETER FILE FOUND')
      STOP 'NO PARAMETER FILE FOUND'
  350 WRITE(6,360)
  360 FORMAT (X'NO DATA FILE FOUND')
      STOP 'NO DATA FILE FOUND'
  370 STOP
      END
```

```
DATAIN  METPLT.FOR      FORTRAN V.5A(621) /KI   [illegible]        8:22    [illegible]

      SUBROUTINE DATAIN(OTITLE)
C
C     SUBROUTINE READS IN DATA SET PARAMETERS, [illegible] CORRESPONDING TO EACH
C     DATA SET, AND THE DATA ARRAY FROM [illegible]
C
      PARAMETER NDATA=300,MCHAN=25
      DIMENSION OTITLE(25)
      COMMON NUMX(MCHAN),NAMEX(2,MCHAN),[illegible],XMIN(MCHAN),
     1 XINC(MCHAN),XMAX(MCHAN),[illegible](MCHAN),[illegible],TYPEX(MCHAN),
     2 NUMY(MCHAN),NAMEY(2,MCHAN),UNITY(2,[illegible]),[illegible](MCHAN),
     3 YINC(MCHAN),YMAX(MCHAN),[illegible](MCHAN),[illegible],TYPEY(MCHAN),
     4 NUMY2(MCHAN),NAMEY2(2,MCHAN),[illegible],[illegible]N2(MCHAN),
     5 YINC2(MCHAN),YMAX2(MCHAN),[illegible](MCHAN),
     6 TYPEY2(MCHAN)
      COMMON /ARRAY/ DATA(NDATA,MCHAN),[illegible],IREF(MCHAN,4)
      COMMON /REAL/ FORM(MCHAN,3),UNAME([illegible])
      COMMON /I1/ IDATE,IYEAR,IDAY,LDATE,[illegible]
      COMMON /N/ NCHAN,NROW,NXTNUM,NDAYS
      DATA DASH,BLANK/'----- ','     '/
C
C     CHECK TO SEE IF A NEW PAGE HEADING [illegible]
C
      ISKIP = 0
      DO 10 I=1,25
   10 IF (OTITLE(I) .NE. BLANK) ISKIP = 1
      IF (ISKIP .EQ. 1) GO TO 30
C
C     USE OLD PAGE HEADING IF A NEW ONE [illegible] READ
C
      READ (7,20) (OTITLE(I),I=1,24)
   20 FORMAT (X24A1)
      GO TO 40
   30 SKIP RECORD 7
C
C     READ PARAMETERS OUTPUT BY THE PREVIOUS [illegible]
C
   40 READ (7,50) NCHAN,NROW,NXTNUM,IYEAR,[illegible],LYEAR,
     1 LDATE,LDAY
   50 FORMAT (3(X13),2(/XI4,X13,XA3))
C
C     READ IN CHANNEL NAMES, UNITS, AND [illegible]
C
      READ (7,60) ((IREF(I,J),J=1,3),(UNA[illegible],J=1,2),(FORM(I,J),
     1 J=1,3),I=1,NCHAN)
   60 FORMAT (XI3,X2A4,X2A4,X3A4)
C
C     READ IN TIME ARRAY
C
      READ (7,70) ((ITIME(I,J),J=1,3),I=1,[illegible])
   70 FORMAT (6(XI4,XI3,XI4))
C
C     READ IN DATA
C
      DO 90 J=1,NCHAN
      SKIP RECORD 7
      READ (7,80) (DATA(I,J),I=1,NROW)
```

```
      80 FORMAT (6(1E12.6))
C
C        LOOK FOR E-FORMAT REPRESENTATION OF "-----" AND REPLACE WITH THE
C        A-FORMAT REPRESENTATION
C
         DO 90 I=1,NROW
      90 IF (DATA(I,J) .EQ. .3758915E+16) DATA([illegible]) = DASH
         RETURN
         END
```



```
         SUBROUTINE INIT(*)
C
C        SUBROUTINE INITIALIZES [illegible], AND DATE TO "FIRST PLOTTED"
C        VALUES.
C
         COMMON /P/ IPDATE,IPYEAR,IPDAY,[illegible]
         COMMON /JN/ JPDATE,JPYEAR,[illegible]
         COMMON /RC/ IROW,KROW,[illegible],ICOL,[illegible]
         COMMON /INTPLT/ IPLOT,[illegible]
         COMMON /DAY/ NAMDAY(7),NUMDAY,[illegible]
C
C        INITIALIZE "END OF LAST PLOT" DATE AND [illegible]
C
         KPYEAR = IPYEAR
         KPDATE = IPDATE - 1
         KPDAY = IPDAY
C
C        INITIALIZE "LAST ROW" [illegible] AND [illegible]
C
         LROW = 1
         NPLOT = 1
C
C        FIND INITIAL PLOT DAY IN DAY NAME [illegible]
C
         NUMDAY = 0
      10 IF (NAMDAY(NUMDAY+1) .EQ. IPDAY) [illegible]
         NUMDAY = NUMDAY + 1
         IF (NUMDAY .LE. 6) GO TO 10
         WRITE (6,20) IPDAY
      20 FORMAT (1X"",A4," IS NOT FOUND [illegible]")
         RETURN 1
         END
```

```
KEYVAL  NETPLT.FOR      FORTRAN V.5A(621) /KI   [illegible]        8:22    PAGE 1

        FUNCTION KEYVAL(WORD)
C
C        FUNCTION CHECKS THE INPUT WORD AND [illegible] AS FOLLOWS:
C
C               WORD          KEYVAL
C               ----          ------
C              "    "           4
C              "YAXIS"          3
C              "XAXIS"          2
C               OTHER           1
C
        DATA XAXIS,YAXIS,BLANK/'XAXIS','YAXIS','     '/
        KEYVAL = 1
        IF (WORD .EQ. XAXIS) KEYVAL = 2
        IF (WORD .EQ. YAXIS) KEYVAL = 3
        IF (WORD .EQ. BLANK) KEYVAL = 4
        RETURN
        END
```

```
IPTYPE  NETPLT.FOR      FORTRAN V.5A(621) /KI   [illegible]        8:22    PAGE 1

        FUNCTION IPTYPE(WORD)
C
C       FUNCTION CHECKS THE INPUT WORD AND ASSIGNS A NUMBER AS FOLLOWS:
C
C               WORD          IPTYPE
C               ----          ------
C              "AVEC "          7
C              "RVEC "          6
C              "YVEC "          5
C              "XVEC "          4
C              "LOG  "          3
C              "LIN  "          2
C               OTHER           1
C
        DATA RLIN,RLOG,XVEC,YVEC,RVEC,AVEC/'LIN  ','LOG  ','XVEC ',
       1 'YVEC ','RVEC ','AVEC '/
        IPTYPE = 1
        IF (WORD .EQ. RLIN) IPTYPE = 2
        IF (WORD .EQ. RLOG) IPTYPE = 3
        IF (WORD .EQ. XVEC) IPTYPE = 4
        IF (WORD .EQ. YVEC) IPTYPE = 5
        IF (WORD .EQ. RVEC) IPTYPE = 6
        IF (WORD .EQ. AVEC) IPTYPE = 7
        RETURN
        END
```

```
XAXIN   NETPLT.FOR      FORTRAN V.5A(621) /KI   [illegible]        8:22    PAGE 1

        SUBROUTINE XAXIN(*)
C
C       SUBROUTINE HANDLES INPUT OF X-AXIS [illegible]
C
        PARAMETER NDATA=300,NCHAN=25
        COMMON NUMX(NCHAN),NAMEX(2,NCHAN),[illegible],XMIN(NCHAN),
       1 XINC(NCHAN),XMAX(NCHAN),[illegible],TYPEX(NCHAN),
       2 NUMY(NCHAN),NAMEY(2,NCHAN),UNITY(2,NCHAN),[illegible],
       3 YINC(NCHAN),YMAX(NCHAN),[illegible],TYPEY(NCHAN),
       4 NUMY2(NCHAN),NAMEY2(2,NCHAN),UNITY2(2,NCHAN),YMIN2(NCHAN),
       5 YINC2(NCHAN),YMAX2(NCHAN),[illegible],
       6 TYPEY2(NCHAN)
        COMMON /INTPLT/ IPLOT,[illegible],NPLOT
C
C       READ X-AXIS PARAMETERS
C
     10 READ (4,20,END=60,ERR=60) NUMX(NPLOT),[illegible],
       1 NAMEX(2,NPLOT),UNITX(1,NPLOT),UNITX(2,NPLOT),XMIN(NPLOT),
       2 XINC(NPLOT),XMAX(NPLOT),[illegible],TYPEX(NPLOT)
     20 FORMAT (XI3,X2A4,X2A4,5(XF5.0),XA5)
        GO TO (30,50,30,30,30,30,30) [illegible]
     30 WRITE (6,40) TYPEX(NPLOT)
     40 FORMAT (X'*',A5,'* IS NOT A VALID X-AXIS [illegible] IS SKIPPED.')
        GO TO 10
     50 RETURN
     60 RETURN 1
        END
```

```
      SUBROUTINE YAXIN(*)
C
C     SUBROUTINE HANDLES INPUT OF Y-AXIS [illegible]
C
      PARAMETER NDATA=300,NCHAN=25
      COMMON NUMX(NCHAN),NAMEX(2,NCHAN),[illegible],XMIN(NCHAN),
     1 XINC(NCHAN),XMAX(NCHAN),[illegible],TYPEX(NCHAN),
     2 NUMY(NCHAN),NAMEY(2,NCHAN),[illegible],YMIN(NCHAN),
     3 YINC(NCHAN),YMAX(NCHAN),[illegible],TYPEY(NCHAN),
     4 NUMY2(NCHAN),NAMEY2(2,NCHAN),[illegible],YMIN2(NCHAN),
     5 YINC2(NCHAN),YMAX2(NCHAN),[illegible],
     6 TYPEY2(NCHAN)
      COMMON /INTPLT/ IPLOT,[illegible]
C
C     READ FIRST Y-AXIS PARAMETERS
C
   10 READ (4,20,END=150,ERR=150) NUMY([illegible]),NAMEY(1,NPLOT),
     1 NAMEY(2,NPLOT),UNITY(1,[illegible]),UNITY(2,[illegible]),YMIN(NPLOT),
     2 YINC(NPLOT),YMAX(NPLOT),[illegible],TYPEY(NPLOT)
   20 FORMAT (I3,X2A4,X2A4,5(F5.0),A5)
C
C     CHECK TYPE OF PLOT REQUESTED
C
      GO TO (30,50,30,30,30,60,90) [illegible]
C
C     TYPE IS UNDEFINED - WRITE ERROR MESSAGE
C
   30 WRITE (6,40) TYPEY(NPLOT)
   40 FORMAT (X"",A5," IS NOT A VALID [illegible] PLOT IS SKIPPED.")
      GO TO 110
C
C     TYPE = "LIN" - LOOK FOR NEXT Y-AXIS
C
   50 CALL XCOPY
      GO TO 110
C
C     TYPE = "BVEC" - LOOK FOR MATCHING [illegible] - "AVEC"
C
   60 READ (4,20,END=150,ERR=150) NUMY2([illegible]),[illegible](1,NPLOT),
     1 NAMEY2(2,NPLOT),UNITY2(1,NPLOT),[illegible],YMIN2(NPLOT),
     2 YINC2(NPLOT),YMAX2(NPLOT),[illegible](NPLOT),
     3 TYPEY2(NPLOT)
C
C     IF TYPE = "AVEC" IS NOT FOUND, WRITE ERROR MESSAGE
C
      IF (IPTYPE(TYPEY2(NPLOT)) .EQ. 7) GO TO 90
      WRITE (6,70) TYPEY(NPLOT),TYPEY2(NPLOT)
   70 FORMAT (X"",A5," AND ",A5," ARE [illegible] TYPES. PLOT",
     1 " SKIPPED.")
      BACKSPACE 4
      GO TO 110
   80 CALL XCOPY
      GO TO 110
C
C     TYPE = "AVEC" - SAVE "AVEC" PARAMETERS AND LOOK FOR MATCHING
C                   TYPE = "BVEC"
```



```
C
   90 NUMY2(NPLOT) = NUMY(NPLOT)
      NAMEY2(1,NPLOT) = NAMEY(1,NPLOT)
      NAMEY2(2,NPLOT) = NAMEY(2,NPLOT)
      UNITY2(1,NPLOT) = UNITY(1,NPLOT)
      UNITY2(2,NPLOT) = UNITY(2,NPLOT)
      YMIN2(NPLOT) = YMIN(NPLOT)
      YINC2(NPLOT) = YINC(NPLOT)
      YMAX2(NPLOT) = YMAX(NPLOT)
      THLDY2(NPLOT) = THLDY(NPLOT)
      DELY2(NPLOT) = DELY(NPLOT)
      TYPEY2(NPLOT) = TYPEY(NPLOT)
      READ (4,20,END=150,ERR=150) NUMY(NPLOT),NAMEY(1,NPLOT),
     1 NAMEY(2,NPLOT),UNITY(1,NPLOT),UNITY(2,NPLOT),YMIN(NPLOT),
     2 YINC(NPLOT),YMAX(NPLOT),THLDY(NPLOT),[illegible](NPLOT),TYPEY(NPLOT)
C
C     IF TYPE = "BVEC" IS NOT FOUND, WRITE ERROR MESSAGE
C
      IF (IPTYPE(TYPEY(NPLOT)) .EQ. 6) GO TO 100
      WRITE (6,70) TYPEY(NPLOT),TYPEY2(NPLOT)
      BACKSPACE 4
      GO TO 110
  100 CALL XCOPY
      GO TO 110
C
C     LOOK FOR MORE Y-AXIS PARAMETERS
C
  110 READ (4,120,END=150) WORD
  120 FORMAT (XA5)
      GO TO (130,140,10,110) KEYVAL(WORD)
  130 BACKSPACE 4
      GO TO 10
  140 BACKSPACE 4
  150 RETURN 1
      END
```

```
MAIN.   METPLT.FOR       FORTRAN V.5A(621) /KI   [illegible]      8:22     PAGE 1

              SUBROUTINE XCOPY
        C
        C     SUBROUTINE COPIES DEFAULT X-AXIS PARAMETERS
        C
              PARAMETER NDATA=300,NCHAN=25
              COMMON NUMX(NCHAN),NAMEX(2,NCHAN),[illegible],XMIN(NCHAN),
             1 XINC(NCHAN),XMAX(NCHAN),THLDA(NCHAN),[illegible],TYPEX(NCHAN),
             2 NUMY(NCHAN),NAMEY(2,NCHAN),UNITY(2,NCHAN),[illegible](NCHAN),
             3 YINC(NCHAN),YMAX(NCHAN),THLDY(NCHAN),[illegible](NCHAN),TYPEY(NCHAN),
             4 NUMY2(NCHAN),NAMEY2(2,NCHAN),UNITY2(2,NCHAN),YMIN2(NCHAN),
             5 YINC2(NCHAN),YMAX2(NCHAN),THLDY2(NCHAN),[illegible]2(NCHAN),
             6 TYPEY2(NCHAN)
              COMMON /INTPLT/ IPLOT,LPLOT,NPLOT
              NUMX(NPLOT+1) = NUMX(NPLOT)
              NAMEX(1,NPLOT+1) = NAMEX(1,NPLOT)
              NAMEX(2,NPLOT+1) = NAMEX(2,NPLOT)
              UNITX(1,NPLOT+1) = UNITX(1,NPLOT)
              UNITX(2,NPLOT+1) = UNITX(2,NPLOT)
              XMIN(NPLOT+1) = XMIN(NPLOT)
              XINC(NPLOT+1) = XINC(NPLOT)
              XMAX(NPLOT+1) = XMAX(NPLOT)
              THLDX(NPLOT+1) = THLDX(NPLOT)
              DELX(NPLOT+1) = DELX(NPLOT)
              TYPEX(NPLOT+1) = TYPEX(NPLOT)
              NPLOT = NPLOT + 1
              RETURN
              END
```

```
SETUP   METPLT.FOR       FORTRAN V.5A(621) /KI   [illegible]      8:22     PAGE 1

              SUBROUTINE SETUP(NSET,TITLPL,*)
        C
        C     SUBROUTINE SETS THE FINAL JULIAN DATE (AND DAY OF THE WEEK) FOR THE
        C     CURRENT PLOTS. ROW NUMBERS FOR THE INITIAL AND FINAL DATES ARE
        C     ALSO FOUND.
        C
              PARAMETER NDATA=300,NCHAN=25
              DIMENSION TITLPL(25)
              COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible](NDATA,3),IPZP(NCHAN,4)
              COMMON /DAY/ NAMDAY(7),NUMDAY,DAYS[illegible]
              COMMON /IL/ IDATE,IYEAR,IDAY,LDATE,[illegible]
              COMMON /N/ NCHAN,NROW,NKINUM,NDAYS
              COMMON /P/ IPDATE,IPYEAR,IPDAY,LPDA[illegible],LPDAY
              COMMON /JK/ JPDATE,JPYEAR,JPDAY,KPDA[illegible],KPDAY
              COMMON /RC/ IROW,KROW,LROW,ICOLX,ICO[illegible]2
              DATA I1,I2,F1,F2 /' INI','FIAL',' [illegible]'/
        C
        C     SET FIRST DATE AND DAY EQUAL TO THE [illegible] DAY AFTER THE END OF THE
        C     PREVIOUS PLOT SET.
        C
              JPDATE = KPDATE + 1
              NUMDAY = NUMDAY + 1
              IF (NUMDAY .GT. 7) NUMDAY = NUMDAY - 7
              JPDAY = NAMDAY(NUMDAY)
              JPYEAR = KPYEAR
              IF ((JPDATE + NDAYS - 1) .GT. LPDATE) NDAYS = LPDATE - JPDATE + 1
        C
        C     SET LAST DATE AND DAY EQUAL TO "NDAYS" AFTER FIRST DATE AND DAY
        C
              KPDATE = JPDATE + NDAYS - 1
              NUMDAY = NUMDAY + NDAYS - ((NDAYS/7)*7) - 1
              IF (NUMDAY .GT. 7) NUMDAY = NUMDAY - 7
              IF (NUMDAY .LT. 1) NUMDAY = NUMDAY + 7
              KPDAY = NAMDAY(NUMDAY)
              KPYEAR = JPYEAR
        C
        C     WRITE PAGE HEADING FOR OUTPUT SUMMARY
        C
              WRITE (6,10) TITLPL,JPDAY,JPDATE,JP[illegible],KPDAY,KPDATE,KPYEAR,NSET
           10 FORMAT (1H1,53X,25A1/40X'DATA FOR [illegible] ',A3,XI3,',',I4,
             1 ' THROUGTH ',A3,XI3,',',I4//X'SUMMARY [illegible] SET NUMBER ',I3//)
        C
        C     FIND THE ROW NUMBER FOR THE BEGINNING [illegible] FIRST JULIAN DATE WHICH IS
        C     TO BE PLOTTED.
        C
              IROW = LROW
           20 IF (ITIME(IROW,2) .GE. JPDATE) GO TO 40
              IROW = IROW + 1
              IF (IROW .LE. NROW) GO TO 20
              WRITE (6,30) I1,I2,JPDATE
           30 FORMAT (1X,2A4,' JULIAN DATE (',I3,') [illegible] FOUND'/
             1 X'PLOT SET IS SKIPPED')
              RETURN 1
        C
        C     FIND THE ROW NUMBER FOR THE END OF [illegible] JULIAN DATE WHICH IS TO BE
        C     PLOTTED.
```

```
LCAD     METPLT.FOR     FORTRAN V.5A(621) /KI    [illegible]     8:22     PAGE 1

      SUBROUTINE LOAD (IPLOT,*)
C
C     SUBROUTINE LOCATES THE APPROPRIATE [illegible] IN THE DATA ARRAY, TESTS
C     THIS DATA AND LOADS THE X- AND Y-ARRAYS [illegible] ACCEPTABLE DATA IS
C     FOUND. IF ICOLX = 0, THEN "ITIME" [illegible] "DATA" IS USED FOR THE X-
C     AXIS.
C
      PARAMETER NDATA=300,NCHAN=25
      DIMENSION TYPE1(2),TYPE2(2)
      REAL NAMEX
      COMMON NUMX(NCHAN),NAMEX(2,NCHAN),[illegible],XMIN(NCHAN),
     1 XINC(NCHAN),XMAX(NCHAN),THLDX(NCHAN),[illegible],TYPEX(NCHAN),
     2 NUMY(NCHAN),NAMEY(2,NCHAN),UNITY(2,[illegible]),[illegible](NCHAN),
     3 YINC(NCHAN),YMAX(NCHAN),THLDY(NCHAN),[illegible](NCHAN),TYPEY(NCHAN),
     4 NUMY2(NCHAN),NAMEY2(2,NCHAN),UNITY2[illegible],YMIN2(NCHAN),
     5 YINC2(NCHAN),YMAX2(NCHAN),THLDY2([illegible]),[illegible](NCHAN),
     6 TYPEY2(NCHAN)
      COMMON /N/ NCHAN,NROW,NATNUM,NDAYS
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible],IFEF(NCHAN,4)
      COMMON /JK/ JPDATE,JPYEAR,JPDAY,[illegible],JADAY
      COMMON /RC/ IROW,KROW,LROW,ICOLX,[illegible]
      COMMON /POINT/ XARRAY(NDATA),YMAG([illegible]),[illegible](NDATA)
      DATA X,Y,Y2,NX,NY,NY2,TYPE1,TYPE2,[illegible]/'X ','Y ','Y2',
     1 'NUMX ','NUMY ','NUMY2','LINEAR    ','[illegible]    ','    ','TIME'/
C
C     FIND DATA MATRIX COLUMN NUMBERS
C
      ICOLX = -1
      ICOLY = -1
      ICOLY2 = -1
      DO 10 I=1,NCHAN
      IF (NUMX(IPLOT) .EQ. [illegible](I,1)) [illegible]
      IF (NUMY(IPLOT) .EQ. [illegible](I,1)) [illegible]
      IF (NUMY2(IPLOT) .EQ. [illegible](I,1)) [illegible]
   10 CONTINUE
      IF (NUMX(IPLOT) .NE. 0) GO TO 20
C
C     IF X-AXIS IS TIME, RESET XMIN,XINC,[illegible], AND [illegible] TO DEFAULTS
C
      ICOLX = 0
      XMIN(IPLOT) = 0
      XINC(IPLOT) = 720
      XMAX(IPLOT) = NDAYS * 1440
      IF ((NAMEX(1,IPLOT) .EQ. BLANK) .AND. ([illegible](IPLOT) .EQ.
     1 BLANK)) NAMEX(1,IPLOT) = [illegible]
C
C     SET ITYPE = 1 FOR LINEAR; [illegible]
C
   20 ITYPE = 1
      KEY = IPTYPE(TYPEY(IPLOT))
      IF ((KEY .EQ. 4) .OR. (KEY .EQ. 6)) [illegible] = 2
C
C     WRITE SUMMARY FOR CURRENT PLOT
C
      WRITE (6,30) IPLOT,(NAMEY(I,IPLOT),[illegible](I,IPLOT),I=1,2)
   30 FORMAT (//X'PLOT NUMBER ',I3,': ',2A5,' [illegible] ',2A5/)
```

```
SETUP    METPLT.FOR     FORTRAN V.5A(621) /K.   [illegible]     8:22     PAGE 1-1

C
   40 LROW = IROW
   50 IF (ITIME(LROW,2) .GT. KPDATE) GO TO [illegible]
      LROW = LROW + 1
      IF (LROW .LE. NROW) GO TO 50
      IF (ITIME(LROW-1,2) .EQ. KPDATE) GO TO [illegible]
      WRITE (6,30) P1,P2,KPDATE
      RETURN 1
   60 LROW = LROW - 1
      RETURN
      END
```

```
C
C     WRITE X-CHANNEL PARAMETERS
C
      IF (DELX(IPLOT) .GE. 0) WRITE (6,40) [illegible],THLDX(IPLOT),
     1 DELX(IPLOT)
      IF (DELX(IPLOT) .LT. 0) WRITE (6,50) [illegible],THLDX(IPLOT)
C
C     WRITE Y-CHANNEL PARAMETERS
C
      IF (DELY(IPLOT) .GE. 0) WRITE (6,40) [illegible],THLDY(IPLOT),
     1 DELY(IPLOT)
      IF (DELY(IPLOT) .LT. 0) WRITE (6,50) [illegible],THLDY(IPLOT)
   40 FORMAT (10X,A2,' CHANNEL = ',I3,'; [illegible] = ',F5.0,
     1 '; HYSTERESIS = ',F5.0)
   50 FORMAT (10X,A2,' CHANNEL = ',I3,'; [illegible] = ',F5.0,
     1 '; HYSTERESIS IS IGNORED')
C
C     CHECK PLOT TYPE
C
      GO TO (60,90) ITYPE
C
C     LINEAR PLOT
C
   60 WRITE (6,70) (TYPE1(I),I=1,2)
   70 FORMAT (10X,'TYPE = ',2A5//)
      GO TO 90
C
C     WRITE Y2-CHANNEL PARAMETERS FOR [illegible]
C
   80 IF (DELY2(IPLOT) .GE. 0) WRITE (6,40) [illegible](IPLOT),
     1 THLDY2(IPLOT),DELY2(IPLOT)
      IF (DELY2(IPLOT) .LT. 0) WRITE (6,50) [illegible](IPLOT),
     1 THLDY2(IPLOT)
      WRITE (6,70) (TYPE2(I),I=1,2)
   90 IF ((ICOLX .LT. 0) .OR. (ICOLY .LT. 0)) GO TO 150
      IF ((ICOLY2 .LT. 0) .AND. ([illegible])) GO TO 150
      KOUNT = 1
      DO 140 KROW=IROW,LROW
      NPLTX = 1
      NPLTY = 1
      NPLTY2 = 1
      IF (ICOLX) 110,100,110
C
C     X-AXIS IS TIME -- CONVERT ELAPSED [illegible]
C
  100 FNEWX = (ITIME(KROW,2) - [illegible]) * [illegible](KROW,3)/100)
     1 * 60 + (ITIME(KROW,3) - ((ITIME(KROW,3)/100) * 100))
      GO TO 120
C
C     X-AXIS IS DATA FROM COLUMN "ICOLX"
C
  110 FNEWX = DATA(KROW,ICOLX)
C
C     TEST FOR ACCEPTABLE X DATA
C
  120 CALL CHECK (1,FNEWX,OLDX,THLDX(IPLOT),[illegible](IPLOT),NPLTX,
```



```
     1 6140)
      FNEWY = DATA(KROW,ICOLY)
C
C     TEST FOR ACCEPTABLE Y DATA (MAGNITUDE)
C
      CALL CHECK (2,FNEWY,OLDY,THLDY(IPLOT),[illegible](IPLOT),NPLTY,
     1 6140)
C
C     SEE IF A VECTOR PLOT WAS REQUESTED
C
      IF (ITYPE .NE. 2) GO TO 130
      FNEWY2 = DATA(KROW,ICOLY2)
C
C     TEST FOR ACCEPTABLE Y2 DATA (DIRECTION)
C
      CALL CHECK (3,FNEWY2,OLDY2,THLDY2([illegible]),[illegible](IPLOT),
     1 NPLTY2,6140)
  130 IF ((NPLTX .EQ. 1) .AND. (NPLTY .EQ. 1) .AND. (NPLTY2 .EQ. 1))
     1 CALL HIST (ITYPE,FNEWX,FNEWY,FNEWY2,[illegible](IPLOT),DELY(IPLOT),
     2 DELY2(IPLOT),6140)
C
C     LOAD ARRAYS IFF ACCEPTABLE DATA WAS [illegible] X AND Y (AND Y2, IN
C     THE CASE OF VECTOR PLOTS). UPDATE "OLD" [illegible] COMPARISON AND
C     INCREMENT ROW COUNTER.
C
      XARRAY(KOUNT) = FNEWX
      OLDX = FNEWX
      YMAG(KOUNT) = FNEWY
      OLDY = FNEWY
      YDIR(KOUNT) = FNEWY2
      OLDY2 = FNEWY2
      KOUNT = KOUNT + 1
  140 CONTINUE
      KROW = KOUNT - 1
      RETURN
C
C     WRITE ERROR MESSAGES IF ANY CHANNEL [illegible] NOT FOUND
C
  150 IF (ICOLX .LT. 0) WRITE (6,160) NX,[illegible]
      IF (ICOLY .LT. 0) WRITE (6,160) NY,[illegible]
      IF ((ICOLY2 .LT. 0) .AND. ([illegible])) WRITE (6,160) NY2,IPLOT
  160 FORMAT (1X,A5,' WAS NOT FOUND IN "[illegible]" -- [illegible] ',
     1 I3,' IS SKIPPED')
      RETURN 1
      END
```

```
CHECK   METPLT.FOR     FORTRAN V.5A(621) /KI   [illegible]        8:22    PAGE 1

      SUBROUTINE CHECK (J,DATNEW,DATOLD,[illegible],NOPLOT,*)
C
C     SUBROUTINE TESTS FOR INVALID DATA ("-----"), VALUES < THRESHOLD,
C     AND VALUES WITHIN A RANGE "DELTA" OF [illegible] VALUE.
C
      PARAMETER NDATA=300,NCHAN=25
      DIMENSION XYZ2(3)
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible],IREF(NCHAN,4)
      COMMON /RC/ IROW,KROW,LROW,ICOLX,[illegible]
      DATA DASH,XYZ2/'----- ',': X =',': Y =',': Z2='/
      IF (DATNEW .NE. DASH) GO TO 20
      WRITE (6,10) (ITIME(KROW,I),I=1,3)
   10 FORMAT (1X14,1X13,1X14,': DATA IS INVALID')
      RETURN 1
   20 IF (DATNEW .GE. THRESH) GO TO 40
      WRITE (6,30) (ITIME(KROW,I),I=1,3),[illegible]
   30 FORMAT (1X14,1X13,1X14,A5,E12.6,' IS < [illegible]')
      RETURN 1
   40 IF (K .EQ. 1) GO TO 60
      IF (DELTA .LT. 0) RETURN
      IF ((DATNEW .LE. (DATOLD - DELTA)) [illegible] .GE.
     1 (DATOLD + DELTA))) GO TO 70
      RETURN
   60 DATOLD = DATNEW
   70 NOPLOT = 0
      RETURN
      END
```

```
HYST    METPLT.FOR     FORTRAN V.5A(621) /KI   [illegible]        8:22    PAGE 1

      SUBROUTINE HYST(ITYPE,[illegible],DELTY,DELTY2,*)
C
C     SUBROUTINE PRINTS [illegible] SUPPRESSED DUE TO
C     FAILURE OF THE HYSTERESIS TEST
C
      PARAMETER NDATA=300,NCHAN=25
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible],IREF(NCHAN,4)
      COMMON /RC/ IROW,KROW,[illegible],ICOLX,[illegible]
      DATA X,Y,Y2/'X = ','Y = ','Y2 = '/
      WRITE (6,10) (ITIME(KROW,I),I=1,3)
   10 FORMAT (1X14,1X13,1X14': POINT SUPPRESSED --')
      IF (DELTX .GE. 0) WRITE (6,20) X,[illegible]
      IF (DELTY .GE. 0) WRITE (6,20) Y,[illegible]
      IF ((ITYPE .EQ. 2) .AND. (DELTY2 [illegible])) WRITE (6,20) Y2,PNEWY2,
     1 DELTY2
   20 FORMAT (1XA5,E12.6,' IS WITHIN ',F5.3,' OF PREVIOUS VALUE')
      WRITE (6,30)
   30 FORMAT (/)
      RETURN 1
      END
```

```
TEXT    METPLT.FOR     FORTRAN V.5A(621) /KI   [illegible]        8:22    PAGE 1

      SUBROUTINE TEXT(ITITLE)
C
C     SUBROUTINE WRITES PAGE TITLE AND SUBTITLE.
C
      PARAMETER NDATA=300,NCHAN=25
      DIMENSION ITITLE(5)
      COMMON /ARRAY/ DATA(NDATA,NCHAN),[illegible],IREF(NCHAN,4)
      COMMON /JK/ JPDATE,JPYEAR,JPDAY,KPDATE,[illegible],KPDAY
      COMMON /RC/ IROW,KROW,LROW,ICOLX,[illegible]
      COMMON /ALPHA/ XSTRNG(9),YSTRNG(9)
C
C     WRITE PAGE TITLE WITH 0.3" CHARACTERS.
C
      CALL HEIGHT(0.3)
      STRNGL = XMESS(ITITLE,100)
      X = (11. - STRNGL)/2.0
      CALL MESSAG(ITITLE,100,X,10.0)
C
C     SET UP STRING FOR SUBTITLE
C
      ENCODE(36,10,XSTRNG) JPDAY,ITIME([illegible]),[illegible](IROW,1),KPDAY,
     1 ITIME(LROW,2),ITIME(LROW,1)
   10 FORMAT (A3,1X13,',',I4,' THROUGH ',[illegible],',',[illegible])
C
C     WRITE SUBTITLE WITH 0.2" CHARACTERS.
C
      CALL HEIGHT(0.2)
      STRNGL = XMESS(XSTRNG,100)
      X = (11. - STRNGL)/2.0
      CALL MESSAG(XSTRNG,100,X,10.3)
      CALL RESET('HEIGHT')
      RETURN
      END
```

```
MAIN.   METPLT.FOR     FORTRAN V.5A(621) /KI   [illegible]        8:22    PAGE 1

      SUBROUTINE TIMES
C
C     SUBROUTINE WRITES THE JULIAN DATES AND A LABEL ON THE TIME AXIS.
C
      COMMON /JK/ JPDATE,JPYEAR,JPDAY,KPDATE,[illegible],KPDAY
      COMMON /DAY/ NARDAY(7),NUMDAY,DAYSIZ
      COMMON /X/ X00,X12,X24,XDATE
      CALL HEIGHT(0.2)
      STRNGL = XMESS('JULIAN DATE',11)
      X = (11. - STRNGL)/2.0
      CALL MESSAG('JULIAN DATE',11,X,0.10)
      CALL RESET('HEIGHT')
      X = XDATE
      DO 20 NDATE=JPDATE,KPDATE
      I = NDATE
      CALL INTNO(I,X,0.5)
      X = X + DAYSIZ
   20 CONTINUE
      RETURN
      END
```

```
EX       METPLT.FOR      FORTRAN V.5A(621) /KI   8-[illegible]       8:22    PAGE 1

         SUBROUTINE EX(I)
C
C        SUBROUTINE WRITES TIMES ON THE X-AXIS
C
         COMMON /X/ X00,X12,X24,XDATE
         COMMON /B/ NCHAN,NROW,NATNUM,NDAYS
         COMMON /DAY/ NAMDAY(7),NUMDAY,DAYSIZ
         XSTART = X00
         XMID = X12
         CALL HEIGHT(0.10)
         DO 20 J=1,NDAYS
         IF ((I .EQ. 1) .OR. (J .NE. 1)) CALL [illegible] (J,XSTART,-0.25)
         CALL INTNO(1200,XMID,-0.25)
         XSTART = XSTART + DAYSIZ
      20 XMID = XMID + DAYSIZ
         CALL INTNO(2400,X24,-0.25)
         CALL RESET('HEIGHT')
         RETURN
         END
```

```
VECPLT   METPLT.FOR      FORTRAN V.5A(621) /KI   8-[illegible]       8:22    PAGE 1

         SUBROUTINE VECPLT (IPLOT,*)
C
C        SUBROUTINE DRAWS VECTORS WITH MAGNITUDES AND DIRECTIONS DEFINED BY
C        YMAG AND YDIR
C
         PARAMETER NDATA=300,NCHAN=25
         COMMON NUMX(NCHAN),NAMEX(2,NCHAN),UNITX(2,NCHAN),XMIN(NCHAN),
        1 XINC(NCHAN),XMAX(NCHAN),THLDX(NCHAN),[illegible](NCHAN),TYPEX(NCHAN),
        2 NUMY(NCHAN),NAMEY(2,NCHAN),UNITY(2,NCHAN),YMIN(NCHAN),
        3 YINC(NCHAN),YMAX(NCHAN),THLDY(NCHAN),[illegible](NCHAN),TYPEY(NCHAN),
        4 NUMY2(NCHAN),NAMEY2(2,NCHAN),UNITY2(2,NCHAN),YMIN2(NCHAN),
        5 YINC2(NCHAN),YMAX2(NCHAN),THLDY2(NCHAN),[illegible](NCHAN),
        6 TYPEY2(NCHAN)
         COMMON /BC/ IROW,NROW,LROW,ICOLX,[illegible],[illegible]
         COMMON /POINT/ XARRAY(NDATA),YMAG(NDATA),YDIR(NDATA)
         COMMON /SIZE/ XSIZE,YSIZE
C

C        CALCULATE SCALE FACTORS IN INCHES/UNIT
C
         YUNITS = YSIZE/(YMAX(IPLOT) - YMIN(IPLOT))
         XUNITS = XSIZE/(XMAX(IPLOT) - XMIN(IPLOT))
C
C        DEFINE AXIS SYSTEM
C
         CALL GPAF(XMIN(IPLOT),XINC(IPLOT),XMAX(IPLOT),YMIN(IPLOT),
        1 YINC(IPLOT),YMAX(IPLOT))
C
C        PLOT AND LABEL A SMALL VECTOR (POINTING UP TO INDICATE ZERO DEGREES)
C
         CALL VECTOR(0.2,2.2,0.2,2.7,1221)
         CALL MESSAG('0',1,0.15,2.75)
C
C        SET UP VECTORS FOR ALL SETS OF DATA POINTS
C
         YTO = YSIZE * 0.5
         RADIAN = 3.14159/180.0
         DO 170 I=1,NROW
C
C        CALCULATE COMPONENTS OF VECTOR
C
         X = YMAG(I)*SIN(YDIR(I)*RADIAN)
         Y = YMAG(I)*COS(YDIR(I)*RADIAN)
C
C        DETERMINE ABSOLUTE LOCATIONS (IN INCHES) FOR VECTOR END POINTS
C
         XTO = XARRAY(I) * XUNITS
         XFROM = XTO + (X * YUNITS)
         YFROM = YTO + (Y * YUNITS)
C
C        DRAW VECTORS
C
     170 CALL VECTOR(XFROM,YFROM,XTO,YTO,1201)
         RETURN 1
         END
```

STIFLT NETPLT.FOR FORTRAN V.[illegible] [illegible] [illegible]

```
      SUBROUTINE STPLT(IFLCT,*)
C
C     SUBROUTINE PLOTS [illegible]
C
      PARAMETER NDATA=300,NCHAN=[illegible]
      COMMON NUMX(NCHAN),NAMEX(2,NCHAN),[illegible],XMIN(NCHAN),
     1 XINC(NCHAN),XMAX(NCHAN),[illegible],TYPEX(NCHAN),
     2 NUMY(NCHAN),NAMEY(2,NCHAN),UNITY(2,[illegible]),[illegible](NCHAN),
     3 YINC(NCHAN),YMAX(NCHAN),[illegible],TYPEY(NCHAN),
     4 NUMY2(NCHAN),NAMEY2(2,NCHAN),UNITY2([illegible]),[illegible]2(NCHAN),
     5 YINC2(NCHAN),YMAX2(NCHAN),[illegible],
     6 TYPEY2(NCHAN)
      COMMON /PC/ INOW,NPCH,[illegible]
      COMMON /POINT/ XARRAY(NDATA),YARAY([illegible]),[illegible](NDATA)
C
C     DEFINE AXIS SYSTEM
C
      CALL GRAF(XMIN(IFLCT),XINC(IFLCT),[illegible],YMIN(IFLCT),
     1 YINC(IFLCT),YMAX(IFLCT))
C
C     DRAW GRAPH
C
      CALL CURVE(XARRAY,YARAY,[illegible],1)
      RETURN 1
      END
```

Appendix D
Listing of Program METINP
(Version 1.2)

MAIN. METINP.FOR FORTRAN V.5A(621) /KI [illegible] 16:26 PAGE 1

```
      PROGRAM METINP
C
C     METINP VERSION 1.2   SEPT.1981
C
C     JOHN HOOVER
C     CODE 4330
C     U.S. NAVAL RESEARCH LABORATORY
C     WASHINGTON, D.C.   20375
C
C
C     PROGRAM INPUTS PARAMETERS FROM THE [illegible] AND PRODUCES A FORMATTED CONTROL
C     FILE (FOR04.DAT) FOR THE METEOR PROGRAM [illegible]
C
C     PROGRAM PARAMETERS:
C     NLINE=MAXIMUM POSSIBLE NUMBER OF LINES OF INSTRUCTIONS FOR ANY ONE
C           PROGRAM. (CURRENTLY,NCHAN=28,[illegible] TO NLINE=25 IN "METSRT")
C
C     KEYWORDS:
C     THE PROGRAM (AND FUNCTION WITHIN A PROGRAM) FOR WHICH A CONTROL FILE IS
C     TO BE CONSTRUCTED IS SELECTED BY USE OF KEYWORDS. CURRENTLY, THE LEGAL
C     KEYWORDS ARE-
C
C     "SORT "=SELECT PROGRAM "METSRT"
C     "CALC "=SELECT PROGRAM "METCLC"
C             "WIND "=FUNCTION TO CALCULATE [illegible] WIND VELOCITY
C             "MOIST"=FUNCTION TO CALCULATE [illegible] VAPOR LOADING
C     "PLOT "=SELECT PROGRAM "METPLT"
C
      DATA SORT,CALC,PLOT,EXIT,YES,NNO/'SORT ','CALC ','PLOT ','EXIT ',
     1 'YES  ','NO   '/
   10 WRITE (5,30)
   30 FORMAT (3(1H1/)X,'PARAMETERS FOR WHICH PROGRAM ',
     1 '(SORT,CALC,PLOT)?: '$)
   40 READ (5,50) CHOICE
   50 FORMAT (A5)
   60 IF (CHOICE .EQ. YES) GO TO 10
      IF (CHOICE .EQ. NNO) GO TO 9999
      IF (CHOICE .EQ. EXIT) GO TO 9999
      IF (CHOICE .EQ. SORT) CALL SORTIN($70)
      IF (CHOICE .EQ. CALC) CALL CALCIN($70,[illegible],CHOICE)
      IF (CHOICE .EQ. PLOT) CALL PLOTIN($70)
      CALL ILEGAL($40,CHOICE)
   70 WRITE (5,80)
   80 FORMAT (3(1H1/)X,'PARAMETERS FOR ANOTHER PROGRAM?: '$)
      GO TO 40
   90 WRITE (5,100)
  100 FORMAT (3(1H1/)X'"SORT " WILL DESTROY [illegible] FILE. PLEASE',
     1 ' CONFIRM: '$)
      GO TO 40
 9999 STOP
      END
```

SORTIN METINP.FOR FORTRAN V.5A(621) /KI [illegible] 16:26 PAGE 1

```
      SUBROUTINE SORTIN(*)
C
C     SUBROUTINE INPUTS 'METSRT' PARAMETERS
C
      PARAMETER NLINE=28
      INTEGER SRTNUM(13),SRTSPN(2,NLINE)
      COMMON INARAY(5,20),LINE(NLINE,15),[illegible],FMTCUT(NLINE,15)
      DIMENSION SRTPMT(3,13),SRTFMT(13),[illegible]
      DATA SRTPMT/'# OF CHANNELS   ','YEAR            ','JULIAN DATE     ',
     1 'DAY OF WEEK     ','YEAR            ','[illegible] DATE      ',
     2 'DAY OF WEEK     ','CHANNEL NUMBER  ','[illegible] NAME      ',
     3 'INPUT UNITS     ','OUTPUT UNITS    ','[illegible] FORMAT    ',
     4 'PRINT SWITCH    '/
      DATA SRTFMT/'I3','I4','[illegible]','A3','I4','[illegible]','A3','I3','2A4','A2',
     1 '2A4','3A4','I1'/
      DATA SRTSPN(1,1),SRTSPN(2,1),SRTSPN(1,2),SRTSPN(2,2),SRTSPN(1,3),
     1 SRTSPN(2,3)/1,1,2,4,5,7/
      DATA SRTNUM,TITLE/8*1,2,1,2,3,1,'SORT '/
      N = NLINE - 3
      DO 10 I=4,NLINE
      SRTSPN(1,I) = 8
      SRTSPN(2,I) = 13
   10 CONTINUE
      WRITE (5,20)
   20 FORMAT (3(1H1/)X'INITIALIZE METSRT PARAMETERS:'/)
C
C     INITIALIZE INARAY
C
      CALL INIT (1,13)
C
C     GET TITLE FOR OUTPUT
C
      CALL PAGE(ITITLE)
C
C     WRITE KEYWORD AND PAGE TITLE
C
      OPEN (UNIT=4,DEVICE='DSK',ACCESS='S[illegible]')
      WRITE (4,25) ITITLE
   25 FORMAT (X'SORT PARAMETERS'/X,5A5)
      CLOSE (UNIT=4,DEVICE='DSK',ACCESS='S[illegible]')
C
C     GET NUMBER OF CHANNELS
C
   30 CALL TTYIN (SRTPMT,SRTFMT,SRTNUM,1,1)
C
C     WRITE ERROR MESSAGE IF TOO MANY CHANNELS [illegible] REQUESTED
C
      IF (INARAY(1,1) .LE. N) GO TO 50
      WRITE (5,40) INARAY(1,1),N
   40 FORMAT (X'NCHAN = ',I3,': EXCEEDS MAX. [illegible] VALUE (',I3,')'/
     1 X'PLEASE ENTER NEW VALUE')
      CALL INIT (1,1)
      GO TO 30
   50 CALL FILE (SRTPMT,SRTNUM,1,1,1)
      WRITE (5,60)
   60 FORMAT (/1X,'INITIAL DATE:'/)
```

```
C
C     GET YEAR, DATE, AND DAY FOR INITIAL DATA SET
C
      CALL TTYIN (SRTFMT,SRTFMT,SRTNUM,2,4)
      CALL FILE (SRTFMT,SRTNUM,2,2,4)
      WRITE (5,70)
   70 FORMAT (/1X,'FINAL DATE:'/)
C
C     GET YEAR, DATE, AND DAY FOR FINAL DATA SET
C
      CALL TTYIN (SRTFMT,SRTFMT,SRTNUM,5,7)
      CALL FILE (SRTFMT,SRTNUM,3,5,7)
      WRITE (5,80)
   80 FORMAT (/1X'CHANNEL VARIABLES:'/)
      DO 90 I=1,INARAY(1,1)
C
C     GET PARAMETERS FOR EACH CHANNEL
C
      CALL TTYIN (SRTFMT,SRTFMT,SRTNUM,8,13)
      CALL FILE (SRTFMT,SRTNUM,3+I,8,13)
C
C     REINITIALIZE INARAY
C
      CALL INIT (8,13)
   90 CONTINUE
      N = INARAY(1,1) + 3
      OPEN (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
      CALL CHECK (SRTFMT,SRTFMT,SRTNUM,SR[illegible],N,[illegible])
      CLOSE (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
      RETURN 1
      END
```



```
      SUBROUTINE CALCIN(*,*,*,CHOICE)
C
C     SUBROUTINE INPUTS 'NETCLC' PARAMETERS
C
      INTEGER CLCNUM(5)
      DIMENSION CLCPMT(3,5),CLCFMT(5),ITITLE(5)
      DATA CLCPMT/'CHANNEL NUMBER ','CHANNEL NAME   ',
     1 'OUTPUT UNITS   ','OUTPUT FORMAT  ','[illegible] SWITCH   '/
      DATA CLCFMT/'I3','2A4','2A4','3A4','[illegible]'/
      DATA CLCNUM/1,2,2,3,1/
      DATA WIND,RMOIST,EXIT,YES,RNO,PLOT,SORT
     1 /'WIND ','MOIST','EXIT','YES  ','NO   ','PLOT ','SORT '/
      CALL INIT(1,13)
C
C     GET TITLE FOR OUTPUT
C
      CALL PAGE(ITITLE)
C
C     WRITE KEYWORD AND PAGE TITLE
C
      OPEN (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
      WRITE (4,5) ITITLE
    5 FORMAT (/X'CALC PARAMETERS'/X,5A5)
      CLOSE (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
   10 WRITE (5,20)
   20 FORMAT (3(1H1/)X'PARAMETERS FOR WHICH FUNCTION',
     1 ' (WIND,MOIST)?: '$)
   30 READ (5,40) CHOICE
   40 FORMAT (A5)
   45 IF (CHOICE .EQ. YES) GO TO 10
      IF (CHOICE .EQ. RNO) RETURN 1
      IF (CHOICE .EQ. EXIT) RETURN 2
      IF (CHOICE .EQ. WIND)
     1  CALL WNDIN(&50,&45,CLCPMT,CLCFMT,CLCNUM,CHOICE)
      IF (CHOICE .EQ. RMOIST)
     1  CALL MSTIN(&50,&45,CLCPMT,CLCFMT,CLCNUM,CHOICE)
      IF (CHOICE .EQ. PLOT) RETURN 2
      IF (CHOICE .EQ. SORT) RETURN 3
      CALL ILEGAL(&30,CHOICE)
   50 WRITE (5,60)
   60 FORMAT (3(1H1/)X,'PARAMETERS FOR ANOTHER FUNCTION?: '$)
      GO TO 30
      END
```

```
WNDIN    WETINP.FOR      FORTRAN V.5A(621) /KI   1[illegible]        16:26    PAGE 1

        SUBROUTINE WNDIN(*,*,WNDPMT,WNDFMT,WNDNUM,CHOICE)
C
C       SUBROUTINE INPUTS 'WIND' PARAMETERS
C
        PARAMETER NLINE=28
        INTEGER WNDNUM(5),WNDSPN(2,NLINE)
        COMMON INARAY(5,20),LINE(NLINE,15),[illegible],PRNOUT(NLINE,15)
        DIMENSION WNDPMT(3,5),WNDFMT(5),HEAD1(3,3),HEAD2(2,2)
        DATA TITLE,HEAD1,HEAD2/'WIND ',' RELATIVE WIND ',
       1 ' ABSOLUTE SHIP ',' ABSOLUTE WIND ','SPEED     ','DIRECTION '/
        DO 5 I=1,NLINE
        WNDSPN(1,I) = 1
        WNDSPN(2,I) = 5
      5 CONTINUE
        OPEN (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
        WRITE (4,10)
     10 FORMAT (/X'WIND SUBROUTINE')
        WRITE (5,20)
     20 FORMAT (3(1H1/)X'INITIALIZE ABSOLUTE AND RELATIVE VELOCITY PARAMETERS:'/)
     30 L = 1
        DO 50 I=1,3
        DO 50 K=1,2
C
C       INITIALIZE INARAY
C
        CALL INIT (1,5)
        WRITE (5,40) (HEAD1(J,I),J=1,3),(HEAD2(J,K),J=1,2)
     40 FORMAT (5A5)
        CALL TTYIN(WNDPMT,WNDFMT,WNDNUM,1,5)
        CALL FILE (WNDPMT,WNDNUM,L,1,5)
        CALL FIX(L)
        L = L + 1
     50 CONTINUE
        CALL CHECK (WNDPMT,WNDFMT,WNDNUM,WNDSPN,6,TITLE)
        WRITE (4,60)
     60 FORMAT (/)
        CALL REPEAT(630,670,TITLE,CHOICE)
        CLOSE (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
        RETURN 1
     70 RETURN 2
        END
```

```
MSTIN    WETINP.FOR      FORTRAN V.5A(621) /KI   1[illegible]        16:26    PAGE 1

        SUBROUTINE MSTIN(*,*,MSTPMT,MSTFMT,MSTNUM,CHOICE)
C
C       SUBROUTINE INPUTS 'MOIST' PARAMETERS
C
        PARAMETER NLINE=28
        DIMENSION MSTNUM(5),HEAD(3,4),MSTSPN(2,NLINE)
        COMMON INARAY(5,20),LINE(NLINE,15),[illegible](NLINE),PRNOUT(NLINE,15)
        REAL MSTPMT(3,5),MSTFMT(5)
        DATA TITLE,HEAD/'MOIST','AIR TEMPERATURE','AIR PRESSURE    ',
       1 'REL. HUMIDITY  ','H2O VAPOR CONC.'/
        DO 5 I=1,NLINE
        MSTSPN(1,I) = 1
        MSTSPN(2,I) = 5
      5 CONTINUE
        OPEN (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
        WRITE (4,10)
     10 FORMAT (/X'MOIST SUBROUTINE')
        WRITE (5,20)
     20 FORMAT (3(1H1/)X'INITIALIZE MOISTURE PARAMETERS:'/)
     30 DO 50 I=1,4
        L = I
C
C       INITIALIZE INARAY
C
        CALL INIT (1,5)
        WRITE (5,40) (HEAD(J,I),J=1,3)
     40 FORMAT (X3A5)
        CALL TTYIN (MSTPMT,MSTFMT,MSTNUM,1,5)
        CALL FILE (MSTPMT,MSTNUM,L,1,5)
        CALL FIX(L)
     50 CONTINUE
        CALL CHECK (MSTPMT,MSTFMT,MSTNUM,MSTSPN,4,TITLE)
        WRITE (4,60)
     60 FORMAT (/)
        CALL REPEAT(630,670,TITLE,CHOICE)
        CLOSE (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
        RETURN 1
     70 RETURN 2
        END
```

```
      SUBROUTINE PLOTIN(*)
C
C     SUBROUTINE INPUTS 'NETPLT' PARAMETERS
C
      PARAMETER BLINE=28
      INTEGER PLTNUM(17),PLTSPN(2,BLINE)
      COMMON INARAY(5,20),LINE(BLINE,15),[illegible](BLINE),PERCUT(BLINE,15)
      DIMENSION PLTPMT(3,17),PLTFMT(17),[illegible]
      DATA PLTPMT/'# OF DAYS/PAGE ','YEAR           ','JULIAN DATE    ',
     1 'DAY OF WEEK    ','YEAR           ','JULIAN DATE    ',
     2 'DAY OF WEEK    ','XAXIS OR YAXIS ','CHANNEL NUMBER ',
     3 'CHANNEL NAME   ','OUTPUT UNITS   ','MINIMUM VALUE  ',
     4 'INCREMENT      ','MAXIMUM VALUE  ','THRESHOLD      ',
     5 'HYSTERESIS     ','AXIS TYPE      '/
      DATA PLTFMT /'I2','I4','I3','A3','I4','I3','A3','A5','I3',
     1 '2A4','2A4',5*'F5.0','A5'/
      DATA PLTSPN(1,1),PLTSPN(2,1),PLTSPN(1,2),PLTSPN(2,2),PLTSPN(1,3),
     1 PLTSPN(2,3),PLTSPN(1,4),PLTSPN(2,4) /1,1,2,4,5,7,8,8/
      DATA PLTNUM,TITLE,IXAXIS,IYAXIS,[illegible],YES,IAVEC,IPVEC
     1  /9*1,2*2,6*1,'PLOT','XAXIS','YAXIS',[illegible],'YES',
     2 'AVEC','PVEC'/
      DO 10 I=4,BLINE
      PLTSPN(1,I) = 9
      PLTSPN(2,I) = 17
   10 CONTINUE
      KVEC = 0
      WRITE (5,20)
   20 FORMAT (3(1H1/)X'INITIALIZE NETPLT PARAMETERS:'/)
C
C     INITIALIZE INARAY
C
      CALL INIT(1,17)
C
C     GET TITLE FOR OUTPUT
C
      CALL PAGE(ITITLE)
C
C     WRITE KEYWORD AND PAGE TITLE
C
      OPEN (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
      WRITE (4,30) ITITLE
   30 FORMAT (X'PLOT PARAMETERS'/X,5A5)
      CLOSE (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
C
C     GET NUMBER OF DAYS PER PLOT SET
C
      CALL TTYIN(PLTPMT,PLTFMT,PLTNUM,1,1)
      CALL FILE(PLTFMT,PLTNUM,1,1,1)
      WRITE (5,40)
   40 FORMAT (/X'INITIAL PLOT DATE:'/)
C
C     GET YEAR, DATE, AND DAY FOR START OF FIRST PLOT SET
C
      CALL TTYIN(PLTPMT,PLTFMT,PLTNUM,2,4)
      CALL FILE(PLTFMT,PLTNUM,2,2,4)
      WRITE (5,50)
```



```
   50 FORMAT (/X'FINAL PLOT DATE:'/)
C
C     GET YEAR,DATE, AND DAY FOR END OF LAST PLOT SET
C
      CALL TTYIN(PLTPMT,PLTFMT,PLTNUM,5,7)
      CALL FILE(PLTFMT,PLTNUM,3,5,7)
      L = 4
C
C     ASK FOR X-AXIS PARAMETERS
C
   60 CALL TTYIN(PLTPMT,PLTFMT,PLTNUM,8,8)
C
C     WRITE ERROR MESSAGE IF X-AXIS IS NOT FIRST
C
      IF (INARAY(1,8) .EQ. IXAXIS) GO TO 80
      WRITE (5,70)
   70 FORMAT (X'XAXIS MUST BE ENTERED FIRST.'/)
      CALL INIT(8,8)
      GO TO 60
   80 CALL FILE(PLTFMT,PLTNUM,L,8,8)
      PLTSPN(1,L) = 8
      PLTSPN(2,L) = 8
      L = L + 1
   90 CALL TTYIN(PLTPMT,PLTFMT,PLTNUM,9,17)
      CALL FILE(PLTFMT,PLTNUM,L,9,17)
      L = L + 1
      CALL INIT(8,17)
  100 CALL TTYIN(PLTPMT,PLTFMT,PLTNUM,8,8)
      IF (INARAY(1,8) .EQ. IYAXIS) GO TO 130
      IF (INARAY(1,8) .EQ. IXAXIS) GO TO 120
      WRITE (5,110) INARAY(1,8)
  110 FORMAT (X'"',A5,'" IS NOT UNDERSTOOD.'/X'PLEASE TRY AGAIN.')
      GO TO 100
  120 L = L - 1
      GO TO 90
  130 CALL FILE(PLTFMT,PLTNUM,L,8,8)
      PLTSPN(1,L) = 8
      PLTSPN(2,L) = 8
      L = L + 1
  140 CALL TTYIN(PLTPMT,PLTFMT,PLTNUM,9,17)
      CALL FILE(PLTFMT,PLTNUM,L,9,17)
      IAXIS = INARAY(1,17)
      L = L + 1
      CALL INIT(8,17)
      IF ((IAXIS .EQ. IAVEC) .OR. (IAXIS .EQ. IPVEC)) KVEC = KVEC + 1
      IF (KVEC .EQ. 1) GO TO 140
      KVEC = 0
      WRITE (5,150)
  150 FORMAT (X'SPECIFICATIONS FOR ANOTHER AXIS? '$)
      READ (5,160) CHOICE
  160 FORMAT (A5)
      IF (CHOICE .EQ. YES) GO TO 170
      IF (CHOICE .EQ. XAXIS) GO TO 100
      IF (CHOICE .EQ. YAXIS) GO TO 140
      L = L - 1
      OPEN (UNIT=4,DEVICE='DSK',ACCESS='APPEND')
```

```
PLOTIN  SETINP.FOR      FORTRAN V.5A(621) /KI   16-JUL-81        16:26    PAGE 1-2

          CALL CHECK(PLTPNT,PLTFMT,PLTNUM,PLTSIZ,...)
          CLOSE(UNIT=4,DEVICE='DSK',ACCESS='APPEND')
          RETURN 1
      170 CALL TTYIN(PLTPNT,PLTFMT,PLTNUM,8,8)
          IF (INARAY(1,8) .EQ. IYAXIS) GO TO 180
          IF (INARAY(1,8) .EQ. IXAXIS) GO TO 30
          WRITE (5,110) INARAY(1,8)
          GO TO 170
      180 INARAY(1,8) = IXAXIS
          GO TO 80
          END
```

```
PAGE    SETINP.FOR      FORTRAN V.5A(621) /KI   16-JUL-81        16:26    PAGE 1

          SUBROUTINE PAGE(ITITLE)
C
C         SUBROUTINE READS IN THE PAGE HEADING FOR THE OUTPUT
C
          PARAMETER MLINE=28
          INTEGER PAGNUM
          COMMON INARAY(5,20),LINE(MLINE,15),LENGTH(MLINE),FRMOUT(MLINE,15)
          DIMENSION PAGPNT(3,1),ITITLE(5)
          DATA PAGPNT,PAGFMT,PAGNUM,STAR/'OUTPUT ......     ','5A5',5,'*'/
       10 CALL TTYIN(PAGPNT,PAGFMT,PAGNUM,1,1)
          CALL FILE(PAGFMT,PAGNUM,1,1,1)
          WRITE (5,20)
       20 FORMAT (X'VERIFY OUTPUT PAGE TITLE (RETURN TO CONTINUE,',
         1 '* TO EDIT)'/)
          WRITE (5,30) (LINE(1,I),I=1,LENGTH(1))
       30 FORMAT (X,5A5,$)
          READ (5,40) CHAR
       40 FORMAT (A1)
          IF (CHAR .EQ. STAR) GO TO 10
          DO 50 I=1,5
       50 ITITLE(I) = LINE(1,I)
          CALL INIT(1,1)
          WRITE (5,60)
       60 FORMAT (//)
          RETURN
          END
```

```
FIX     SETINP.FOR      FORTRAN V.5A(621) /KI   16-JUL-81        16:26    PAGE 1

          SUBROUTINE FIX(MLINE)
C
C         SUBROUTINE CHECKS CHANNEL NUMBER AND PHASE SWITCH AND SETS THEM TO ZERO
C         IF A BLANK WAS PRESENT
C
          PARAMETER MLINE=28
          COMMON INARAY(5,20),LINE(MLINE,15),LENGTH(MLINE),FRMOUT(MLINE,15)
          DATA IBLANK/'     '/
          IF (LINE(MLINE,1) .EQ. IBLANK) LINE(MLINE,1) = 0
          IF (LINE(MLINE,9) .EQ. IBLANK) LINE(MLINE,9) = 0
          RETURN
          END
```

```
ILEGAL  SETINP.FOR      FORTRAN V.5A(621) /KI   16-JUL-81        16:26    PAGE 1

          SUBROUTINE ILEGAL(*,WORD)
C
C         SUBROUTINE WRITES AN 'ILLEGAL CHOICE' MESSAGE
C
          WRITE (5,10) WORD
       10 FORMAT (1X,'"',A5,'" IS NOT ALLOWED.'/1X,'PLEASE TRY AGAIN OR',
         1 ' TYPE "EXIT" TO QUIT: '$)
          RETURN 1
          END
```

```
REPEAT  SETINP.FOR      FORTRAN V.5A(621) /KI   16-JUL-81        16:26    PAGE 1

          SUBROUTINE REPEAT (*,*,WORD,CHOICE)
C
C         SUBROUTINE ASKS WHETHER THE PREVIOUS PARAMETER INPUT SHOULD BE REPEATED
C
          DATA EXIT,YES,RNO,PLOT,SORT
         1 /'EXIT ','YES  ','NO   ','PLOT ','SORT '/
          WRITE (5,10) WORD
       10 FORMAT (/X'REPEAT PARAMETER INPUT FOR SUBROUTINE "',A5,'"?: '$)
       20 READ (5,30) CHOICE
       30 FORMAT (A5)
          IF (CHOICE .EQ. YES) GO TO 40
          IF (CHOICE .EQ. RNO) RETURN
          IF (CHOICE .EQ. EXIT) RETURN 2
          IF (CHOICE .EQ. PLOT) RETURN 2
          IF (CHOICE .EQ. SORT) RETURN 2
          CALL ILEGAL (&20,CHOICE)
       40 WRITE (5,50)
       50 FORMAT (//)
          RETURN 1
          END
```

```
TTYIN    METINP.FOR      FORTRAN V.5A(621) /KI   [illegible]        16:26    PAGE 1

        SUBROUTINE TTYIN (PROMPT,FORMIN,NUM,[illegible],[illegible])
C
C       SUBROUTINE WRITES [illegible] MESSAGES [illegible] ARRAY 'PROMPT' AND
C       READS RESPONSES FROM THE TTY ACCORDING TO 'FORMIN'
C
        PARAMETER NLINE=28
        COMMON INARAY(5,20),LINE(NLINE,15),[illegible],FRMOUT(NLINE,15)
        DIMENSION PROMPT(3,20),FORMIN(20),NUM(20)
        DIMENSION STRING(5),FORM(3)
        DATA STAR,NLEFT,PLUS,RIGHT/'*    ','(','(1X,',')'/
        DATA DOLLAR,BLANK,IBLANK/'$)',' ','    '/
C
C       SET POINTER, # OF CHARACTERS, AND # OF [illegible]
C
        IPOINT = NFIRST
     10 NUMBER = 5 * NUM(IPOINT)
C
C       SET UP OUTPUT FORMAT
C
        FORM(1) = PLUS
        FORM(2) = FORMIN(IPOINT)
        FORM(3) = DOLLAR
        WRITE (5,20) (PROMPT(I,IPOINT),I=1,3),[illegible],FORM(2),RIGHT
     20 FORMAT (1X,3A5,A1,A3,A1,'=',$)
C
C       WRITE PREVIOUS VALUE, IF ANY.  OTHERWISE, GO TO INPUT.
C
        EMPTY = 1
        DO 25 I=1,NUM(IPOINT)
     25 IF (INARAY(I,IPOINT) .NE. IBLANK) EMPTY = 0
        IF (EMPTY .EQ. 1) GO TO 40
        WRITE (5,FORM) (INARAY(I,IPOINT),I=1,NUM(IPOINT))
        WRITE (5,30)
     30 FORMAT ('+? '$)
C
C       SET UP FORMAT TO DECODE DATA
C
     40 FORM(3) = RIGHT
        FORM(1) = NLEFT
C
C       INPUT RESPONSE
C
        READ (5,50) (STRING(I),I=1,5)
     50 FORMAT (5A5)
C
C       TEST FIRST CHARACTER FOR '*'
C
        DECODE (1,60,STRING) CHAR
     60 FORMAT (A1)
        IF (CHAR .NE. STAR) GO TO 70
C
C       CHARACTER IS A '*', THEREFORE BACK UP TO PREVIOUS INPUT & REPEAT
C
        IPOINT = IPOINT - 1
        IF (IPOINT .LT. NFIRST) IPOINT = NFIRST
        GO TO 10
```

```
TTYIN    METINP.FOR      FORTRAN V.5A(621) /KI   [illegible]        16:26    PAGE 1-1

C
C       CHARACTER IS NOT '*', SO DECODE VALUE FOR CURRENT VARIABLE
C
     70 DECODE (5,65,STRING) CHAR
     65 FORMAT (A5)
        IF (CHAR .EQ. BLANK) GO TO 75
        DECODE (NUMBER,FORM,STRING)(INARAY([illegible]),I=1,NUM(IPOINT))
     75 IF (IPOINT .GE. NLAST) GO TO 80
        IPOINT = IPOINT + 1
        GO TO 10
     80 WRITE (5,90)
     90 FORMAT (/)
        RETURN
        END
```

```
FILE     METINP.FOR      FORTRAN V.5A(621) /KI   [illegible]        16:26    PAGE 1

        SUBROUTINE FILE (FORMIN,NUM,NLINE,[illegible],[illegible])
C
C       SUBROUTINE COMPOSES A LINE AND THE CORRESPONDING FORMAT FOR THE OUTPUT FILE
C
        PARAMETER NLINE=28
        COMMON INARAY(5,20),LINE(NLINE,15),[illegible],FRMOUT(NLINE,15)
        DIMENSION FORMIN(20),NUM(20)
        I = 1
        J = 1
        DO 20 IPOINT=NFIRST,NLAST
        DO 10 IWORD=1,NUM(IPOINT)
        LINE(NLINE,I) = INARAY(IWORD,IPOINT)
        I = I + 1
     10 CONTINUE
        FRMOUT(NLINE,J) = FORMIN(IPOINT)
        J = J + 1
     20 CONTINUE
        LENGTH(NLINE) = I - 1
        RETURN
        END
```

```
CHECK    RETINP.FOR     FORTRAN V.5A(621) /KI   1[illegible]     16:26    PAGE 1

          SUBROUTINE CHECK (PROMPT,FORMIN,NUM,[illegible],LAST,WORD)
    C
    C     SUBROUTINE WRITES OUTPUT ON TTY LINE [illegible] AND WRITES VERIFIED LINES TO THE DISK (FOR04.DAT)
    C
          PARAMETER NLINE=28
          COMMON INARAY(5,20),LINE(NLINE,15),[illegible](NLINE),PRMOUT(NLINE,15)
          DIMENSION PROMPT(3,20),FORMIN(20),NUM(20)
          DIMENSION OUT(22),ISPAN(2,NLINE)
          DATA STAR,OUT(1),X,RIGHT,DOLLAR/'*    ','(1X,',',1X',')','$)'/
          WRITE (5,5)
        5 FORMAT (1X,'VERIFY FILE (TYPE RETURN [illegible], * TO EDIT)'/)
          WRITE (5,10) WORD
       10 FORMAT (1X,A5,' PARAMETERS')
          DO 40 NLINE=1,LAST
          LINNUM = NLINE
          N = LENGTH(NLINE)
          L = 1
          M = 2
       15 IF (PRMOUT(NLINE,L) .EQ. 0) GO TO 20
          OUT(M) = PRMOUT(NLINE,L)
          OUT(M+1) = X
          L = L + 1
          M = M + 2
          GO TO 15
       20 OUT(M) = DOLLAR
       25 WRITE (5,OUT) (LINE(NLINE,L),L=1,N)
          READ (5,30) CHAR
       30 FORMAT (A1)
          IF (CHAR .EQ. STAR) CALL EDIT (PROMPT,[illegible],NUM,LINNUP,ISPAN,&25)
          OUT(M) = RIGHT
          WRITE (4,OUT) (LINE(NLINE,L),L=1,N)
       40 CONTINUE
          RETURN
          END
```

```
EDIT     RETINP.FOR     FORTRAN V.5A(621) /KI   1[illegible]     16:26    PAGE 1

          SUBROUTINE EDIT (PROMPT,FORMIN,NUM,[illegible],*)
    C
    C     SUBROUTINE ALLOWS CORRECTION OF [illegible] PARAMETER FILE
    C
          PARAMETER NLINE=28
          COMMON INARAY(5,20),LINE(NLINE,15),[illegible](NLINE),PRMOUT(NLINE,15)
          DIMENSION PROMPT(3,20),FORMIN(20),NUM(20),ISPAN(2,NLINE)
          WRITE (5,10) N
       10 FORMAT (1H0,X'EDIT LINE # ',I2,':'/)
          J = ISPAN(1,N)
          K = ISPAN(2,N)
       20 CALL RESTOR(NUM,N,J,K)
          CALL TTYIN (PROMPT,FORMIN,NUM,J,K)
          CALL FILE (FORMIN,NUM,N,J,K)
          RETURN 1
          END
```

```
INIT     RETINP.FOR     FORTRAN V.5A(621) /KI   1[illegible]     16:26    PAGE 1

          SUBROUTINE INIT (NFIRST,NLAST)
    C
    C     SUBROUTINE INITIALIZES INPUT ARRAY: [illegible](I,J) = '     '
    C
          PARAMETER NLINE=28
          COMMON INARAY(5,20),LINE(NLINE,15),[illegible](NLINE),PRMOUT(NLINE,15)
          DATA IBLANK/'     '/
          DO 10 J=NFIRST,NLAST
          DO 10 I=1,5
          INARAY(I,J) = IBLANK
       10 CONTINUE
          RETURN
          END
```

```
RESTOR   RETINP.FOR     FORTRAN V.5A(621) /KI   1[illegible]     16:26    PAGE 1

          SUBROUTINE RESTOR (NUM,NLINE,NFIRST,[illegible])
    C
    C     SUBROUTINE RESTORES VALUES TO INARAY
    C
          PARAMETER NLINE=28
          COMMON INARAY(5,20),LINE(NLINE,15),[illegible](NLINE),PRMOUT(NLINE,15)
          DIMENSION NUM(20)
          I = 1
          DO 10 IPOINT=NFIRST,NLAST
          DO 10 IWORD=1,NUM(IPOINT)
          INARAY(IWORD,IPOINT) = LINE(NLINE,I)
          I = I + 1
       10 CONTINUE
          RETURN
          END
```

Appendix E
METINP Terminal Session
Creation of a Parameter File

```
PARAMETERS FOR WHICH PROGRAM (SORT,CALC,PLOT)?: SORT.


INITIALIZE NETSET PARAMETERS:

OUTPUT TITLE    (5A5)=MATES 1981 CRUISE

VERIFY OUTPUT PAGE TITLE (RETURN TO CONTINUE,* TO ED..)

MATES 1981 CRUISE


# OF CHANNELS   (I3 )= 10


INITIAL DATE:

YEAR            (I4 )=1981
JULIAN DATE     (I3 )= 29
DAY OF WEEK     (A3 )=19/91/THU


FINAL DATE:

YEAR            (I4 )=1981
JULIAN DATE     (I3 )= 29
DAY OF WEEK     (A3 )=THU


CHANNEL VARIABLES:

CHANNEL NUMBER (I3 )= 40
CHANNEL NAME   (2A4)=OUT.TEMP
INPUT UNITS    (A2 )= C
OUTPUT UNITS   (2A4)= DEG C
OUTPUT FORMAT  (3A4)=2X,F6.1,2X,
PRINT SWITCH   (I1 )=1

CHANNEL NUMBER (I3 )= 27
CHANNEL NAME   (2A4)=PWR.SUP.
INPUT UNITS    (A2 )= V
OUTPUT UNITS   (2A4)= VOLTS
OUTPUT FORMAT  (3A4)=2X,F7.3,1X,
PRINT SWITCH   (I1 )=0

CHANNEL NUMBER (I3 )= 26
CHANNEL NAME   (2A4)=POS.PWR.
INPUT UNITS    (A2 )= V
OUTPUT UNITS   (2A4)= VOLTS
OUTPUT FORMAT  (3A4)=2X,F7.3,2/2/1X,
PRINT SWITCH   (I1 )=0

CHANNEL NUMBER (I3 )= 22
CHANNEL NAME   (2A4)=OUT.TEMP
INPUT UNITS    (A2 )= V
OUTPUT UNITS   (2A4)= DEG C
OUTPUT FORMAT  (3A4)=2X,F6.1,2X,
PRINT SWITCH   (I1 )=1

CHANNEL NUMBER (I3 )= 21
CHANNEL NAME   (2A4)=REL./ /HUM.
INPUT UNITS    (A2 )= V
OUTPUT UNITS   (2A4)=PER CENT
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,
PRINT SWITCH   (I1 )=0

CHANNEL NUMBER (I3 )= 20
CHANNEL NAME   (2A4)=PRESSURE
INPUT UNITS    (A2 )=MV
OUTPUT UNITS   (2A4)= TORR
OUTPUT FORMAT  (3A4)=2X,F6.1,2X,
PRINT SWITCH   (I1 )=0

CHANNEL NUMBER (I3 )= 15
CHANNEL NAME   (2A4)=SHP.HEAD
INPUT UNITS    (A2 )=MV
OUTPUT UNITS   (2A4)= DEGREES
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,
PRINT SWITCH   (I1 )=1

CHANNEL NUMBER (I3 )= 14
CHANNEL NAME   (2A4)=SHP.SPD.
INPUT UNITS    (A2 )=MV
OUTPUT UNITS   (2A4)= KNOTS
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,
PRINT SWITCH   (I1 )=1

CHANNEL NUMBER (I3 )= 13
CHANNEL NAME   (2A4)=WND.DIR.
INPUT UNITS    (A2 )= V
OUTPUT UNITS   (2A4)= DEGREES
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,
PRINT SWITCH   (I1 )=1

CHANNEL NUMBER (I3 )= 12
CHANNEL NAME   (2A4)=WND.SPD.
INPUT UNITS    (A2 )= V
OUTPUT UNITS   (2A4)= KNOTS
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,
PRINT SWITCH   (I1 )=1

VERIFY FILE (TYPE RETURN TO PROCEED, * TO EDIT)

SORT  PARAMETERS
 10
1981  29 THU
1981  29 THU

40 OUT.TEMP   C   DEG C    2X,F6.1,2X,  1
27 PWR.SUP.   V   VOLTS    2X,F7.3,1X,  0
26 POS.PWR.   V   VOLTS    2X,F7.3,1X,  0
22 OUT.TEMP   V   DEG C    2X,F6.1,2X,  1
21 REL.HUM.   V  PER CENT  2X,F6.0,2X,  0
20 PRESSURE  MV   TORR     2X,F6.1,2X,  0
15 SHP.HEAD  MV   DEGREES  2X,F6.0,2X,  1
14 SHP.SPD.  MV   KNOTS    2X,F6.0,2X,  1
13 WND.DIR.   V   DEGREES  2X,F6.0,2X,  1
12 WND.SPD.   V   KNOTS    2X,F6.0,2X,  1
```

```
PARAMETERS FOR ANOTHER PROGRAM?: YES_


PARAMETERS FOR WHICH PROGRAM (SORT,CALC,PLOT)?: CALC_
OUTPUT TITLE   (5A5)=_

VERIFY OUTPUT PAGE TITLE (RETURN TO CONTINUE,* TO EDIT)
                          _


PARAMETERS FOR WHICH FUNCTION (WIND,HOIST)?: WIND_
```

```
INITIALIZE ABSOLUTE WIND VELOCITY PARAMETERS:

RELATIVE WIND SPEED
CHANNEL NUMBER (I3 )= 12_
CHANNEL NAME   (2A4)=REL.SPD._
OUTPUT UNITS   (2A4)= KNOTS_
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,_
PRINT SWITCH   (I1 )=0_

RELATIVE WIND DIRECTION
CHANNEL NUMBER (I3 )= 13_
CHANNEL NAME   (2A4)=REL.DIR._
OUTPUT UNITS   (2A4)= DEGREES_
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,_
PRINT SWITCH   (I1 )=0_

ABSOLUTE SHIP SPEED
CHANNEL NUMBER (I3 )= 14_
CHANNEL NAME   (2A4)=SHP.SPD._
OUTPUT UNITS   (2A4)= KNOTS_
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,_
PRINT SWITCH   (I1 )=0_

ABSOLUTE SHIP DIRECTION
CHANNEL NUMBER (I3 )= 15_
CHANNEL NAME   (2A4)=SHP.HEAD_
OUTPUT UNITS   (2A4)= DEGREES_
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,_
PRINT SWITCH   (I1 )=0_

ABSOLUTE WIND SPEED
CHANNEL NUMBER (I3 )=100_
CHANNEL NAME   (2A4)=ABS.SPD._
OUTPUT UNITS   (2A4)= KNOTS_
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,_
PRINT SWITCH   (I1 )=1_

ABSOLUTE WIND DIRECTION
CHANNEL NUMBER (I3 )=101_
CHANNEL NAME   (2A4)=ABS.DIR._
OUTPUT UNITS   (2A4)= DEGREES_
OUTPUT FORMAT  (3A4)=2X,F6.0,2X,_
PRINT SWITCH   (I1 )=1_

VERIFY FILE (TYPE RETURN TO PROCEED, * TO EDIT)

WIND  PARAMETERS
 12 REL.SPD.  KNOTS   2X,F6.0,2X,  0 _
 13 REL.DIR.  DEGREES 2X,F6.0,2X,  0 _
 14 SHP.SPD.  KNOTS   2X,F6.0,2X,  0 _
 15 SHP.HEAD  DEGREES 2X,F6.0,2X,  0 _
100 ABS.SPD.  KNOTS   2X,F6.0,2X,  1 _
101 ABS.DIR.  DEGREES 2X,F6.0,2X,  1 _
```

```
REPEAT PARAMETER INPUT FOR SUBROUTINE "WIND "?: NO

PARAMETERS FOR ANOTHER FUNCTION?: MOIST

INITIALIZE MOISTURE PARAMETERS:

AIR TEMPERATURE
CHANNEL NUMBER (I3 )= 40
CHANNEL NAME   (2A4) =OUT.TEMP
OUTPUT UNITS   (2A4) = DEG C
OUTPUT FORMAT  (3A4) =2X,F6.1,2X,
PRINT SWITCH   (I1 ) =0

AIR PRESSURE
CHANNEL NUMBER (I3 )= 20
CHANNEL NAME   (2A4) =PRESSURE
OUTPUT UNITS   (2A4) = TORR
OUTPUT FORMAT  (3A4) =2X,F6.1,2X,
PRINT SWITCH   (I1 ) =0

REL. HUMIDITY
CHANNEL NUMBER (I3 )= 21
CHANNEL NAME   (2A4) =REL.HUM.
OUTPUT UNITS   (2A4) =PER CENT
OUTPUT FORMAT  (3A4) =2X,F6.0,2X,
PRINT SWITCH   (I1 ) =0

H2O VAPOR CONC.
CHANNEL NUMBER (I3 )=102
CHANNEL NAME   (2A4) =H2O VAP
OUTPUT UNITS   (2A4) =  PPMV
OUTPUT FORMAT  (3A4) =2X,F6.0,2X,
PRINT SWITCH   (I1 ) =1

VERIFY FILE (TYPE RETURN TO PROCEED, * TO EDIT)

MOIST PARAMETERS
 40 OUT.TEMP  DEG C    2X,F6.1,2X,  0
 20 PRESSURE  TORR     2X,F6.1,2X,  0
 21 REL.HUM.  PER CENT 2X,F6.0,2X,  0
102 H2O VAP   PPMV     2X,F6.0,2X,  1

REPEAT PARAMETER INPUT FOR SUBROUTINE "MOIST"?: YES

AIR TEMPERATURE
CHANNEL NUMBER (I3 )= 22
CHANNEL NAME   (2A4) =OUT.TEMP
OUTPUT UNITS   (2A4) = DEG C
OUTPUT FORMAT  (3A4) =2X,F6.1,2X,
PRINT SWITCH   (I1 ) =0

AIR PRESSURE
CHANNEL NUMBER (I3 )= 20
CHANNEL NAME   (2A4) =PRESSURE
OUTPUT UNITS   (2A4) =  TORR
OUTPUT FORMAT  (3A4) =2X,F6.1,2X,
PRINT SWITCH   (I1 ) =1
```

```
REL. HUMIDITY
CHANNEL NUMBER (I3 )=_21_
CHANNEL NAME   (2A4)=REL.HUM._
OUTPUT UNITS   (2A4)=PER CENT_
OUTPUT FORMAT  (3A4)=2X,F6.0,2X._
PRINT SWITCH   (I1 )=1_

H2O VAPOR CONC.
CHANNEL NUMBER (I3 )=_
CHANNEL NAME   (2A4)=_
OUTPUT UNITS   (2A4)=_
OUTPUT FORMAT  (3A4)=_
PRINT SWITCH   (I1 )=1_

VERIFY FILE (TYPE RETURN TO PROCEED, * TO EDIT)

HOIST PARAMETERS
 22 OUT.TEMP  DEG C    2X,F6.1,2X,  0 _
 20 PRESSURE   TORR    2X,F6.1,2X,  1 _
 21 REL.HUM. PER CENT 2X,F6.0,2X,   1 _
  0                                 1 _

REPEAT PARAMETER INPUT FOR SUBROUTINE "HOIST"?: NO_


PARAMETERS FOR ANOTHER FUNCTION?: PLOT_
```

```
INITIALIZE HEIPLT PARAMETERS:

OUTPUT TITLE   (5A5)=_

VERIFY OUTPUT PAGE TITLE (RETURN TO CONTINUE,* TO EDIT)


# OF DAYS/PAGE (I2 )=_1_

INITIAL PLOT DATE:

YEAR           (I4 )=1981_
JULIAN DATE    (I3 )=_29_
DAY OF WEEK    (A3 )=THU_

FINAL PLOT DATE:

YEAR           (I4 )=1981_
JULIAN DATE    (I3 )=_29_
DAY OF WEEK    (A3 )=THU_

XAXIS OR YAXIS (A5 )=YAXIS_

XAXIS MUST BE ENTERED FIRST

XAXIS OR YAXIS (A5 )=XAXIS_

CHANNEL NUMBER (I3 )=0_
CHANNEL NAME   (2A4)=_
OUTPUT UNITS   (2A4)=_
MINIMUM VALUE  (F5.)=0._
INCREMENT      (F5.)=0._
MAXIMUM VALUE  (F5.)=0._
THRESHOLD      (F5.)=0._
HYSTERESIS     (F5.)=100._
AXIS TYPE      (A5 )=LIN_

XAXIS OR YAXIS (A5 )=YAXIS_

CHANNEL NUMBER (I3 )=100_
CHANNEL NAME   (2A4)=WND.VEC._
OUTPUT UNITS   (2A4)=_KNOTS_
MINIMUM VALUE  (F5.)=0._
INCREMENT      (F5.)=5._
MAXIMUM VALUE  (F5.)=40._
THRESHOLD      (F5.)=2._
HYSTERESIS     (F5.)=-1._
AXIS TYPE      (A5 )=VEC_
```

```
CHANNEL NUMBER (I3 )=101
CHANNEL NAME   (2A4)=ABS.DIR.
OUTPUT UNITS   (2A4)= DEGREES
MINIMUM VALUE  (F5.)=0.
INCREMENT      (F5.)=45.
MAXIMUM VALUE  (F5.)=360.
THRESHOLD      (F5.)=0.
HYSTERESIS     (F5.)=30.
AXIS TYPE      (A5 )=AYES

SPECIFICATIONS FOR ANOTHER AXIS?: XAXIS
CHANNEL NUMBER (I3 )=0
CHANNEL NAME   (2A4)=
OUTPUT UNITS   (2A4)=
MINIMUM VALUE  (F5.)=0.
INCREMENT      (F5.)=0.
MAXIMUM VALUE  (F5.)=0.
THRESHOLD      (F5.)=0.
HYSTERESIS     (F5.)=0.
AXIS TYPE      (A5 )=LIN

XAXIS OR YAXIS (A5 )=YAXIS

CHANNEL NUMBER (I3 )=100
CHANNEL NAME   (2A4)=ABS.SPD.
OUTPUT UNITS   (2A4)=KNOTS
MINIMUM VALUE  (F5.)=0.
INCREMENT      (F5.)=5.
MAXIMUM VALUE  (F5.)=90.
THRESHOLD      (F5.)=0.
HYSTERESIS     (F5.)=-1.
AXIS TYPE      (A5 )=LIN

SPECIFICATIONS FOR ANOTHER AXIS?: YAXIS
CHANNEL NUMBER (I3 )=101
CHANNEL NAME   (2A4)=ABS.DIR.
OUTPUT UNITS   (  )=
CHANNEL NAME       )=ABS,DIR.? ABS.DIR.
OUTPUT UNITS        = DEGREES
MINIMUM VALUE    5.,=0.
INCREMENT        F5.)=45.
MAXIMUM VALUE  (
THRESHOLD      (
HYSTERESIS
AXIS TYPE        5

SPECIFICATIONS FOR ANOTHER AXIS?: NO
VERIFY FILE (TYPE RETURN TO PROCEED, * TO EDIT)

PLOT  PARAMETERS
 1
1581  29 THU
1581  29 THU
XAXIS
  0                       0.    0.    0.    0.  180.
YAXIS
100 WND.VEC.  KNOTS       0.    5.   40.    2.   -1.
101 ABS.DIR.  DEGREES     0.   45.  360.    0.   30.
XAXIS
  0                       0.    0.    0.    0.    0.
YAXIS
100 ABS.SPD. KNOTS        0.    5.   40.    0.   -1.
101 ABS.DIR.  DEGREES     0.   45.  360.    0.   -1.
```

Appendix F
METSRT Error Messages

Several references to the error detection capabilities of METSRT have previously been made. In this section, these capabilities will be discussed in more detail and some examples will be give.

Five different classes of errors may be distinguished, as follows:

1) Failure to find some piece of required information.
2) Errors occurring during data input.
3) Conditions which were detected and flagged by the data logger.
4) Inability to properly process some type of data.
5) Errors arising from the processing of a specific piece of data.

Errors of the first type are detected in the main program when an attempt to OPEN a file fails or when the keyword "SORT" cannot be found in the parameter file. In all cases, execution is immediately aborted, and the reason is printed out.

Subroutines SEARCH and DATAIN look for errors of the second type, which include such things as read errors, duplicate channel numbers, and incorrectly formatted data records. Each such problem is listed and, insofar as possible, analysis of the remaining data continues.

Once a record has been read and found to be valid, it is checked by FLAG to determine if any of the data logger warning flags were set. Broken or overloaded sensors are detected at this point and these data are ignored. If the data logger's preset upper or lower limits were exceeded, this will also be be detected but, in these cases, the data is not rejected. Finally, if an illegal character appears in one of the positions reserved for data logger flags, this fact will be reported.

In general, each type of data will require a special function subprogram to perform the appropriate calculations and conversion of units. MANIP uses a computed GO TO statement to select the correct function for each channel. In the event that the channel number lies outside the range of the GO TO, or if no function has been defined at the specified statement label, then an error message will be printed. The data will be left in its original (input) form.

During processing of a specific data record, various errors may occur. For the most part, it is left up to the individual function subprograms to make any tests which may be required. For example, the values of various power supply voltages are tested and messages printed if they lie outside the specified ranges.

One general requirement of all data, regardless of type, is that it must have the correct units (those expected by the corresponding subprogram). For each channel, the proper units are declared in the parameter file at input time. DIDDLE compares the actual units with the expected units and lists all

discrepancies.

Most of the error messages presently incorporated in METSRT are illustrated in the following three tables. Table F1 shows the complete error listing as it appears in FORØ6.DAT. In Table F2, those messages produced during data input are related to the errors in the data file which caused them. Table F3 similarly correlates processing time error messages with the erroneous data records.

In all of these examples, the parameter and data files were specifically constructed to exercise the maximum possible number of error trapping routines. Not illustrated are those (such as "NO DATA FILE FOUND") which are mutually exclusive.

Table F1
METSRT Error Messages

```
                                                    [illegible] MESSAGE DEMO.
                                                         WARNING!

1979 307 2200 CHANNEL 18 : LOWER LIMIT EXCEEDED.
1979 307 2200 CHANNEL 22 : SENSOR OVERLOAD
1979 307 2200 CHANNEL 23 : UNEXPECTED CHARACTER : [illegible]
1979 307 2200 CHANNEL 24 : SENSOR BROKEN
INCOMPREHENSIBLE DATA AT 2200 HOURS ON DAY 307,1979:
A BAD LINE
SYSTEM RESET AFTER 2200 HOURS ON DAY 307,1979. DATA SET(S) SKIPPED.
ACQUISITION RESUMED AT 2230 HOURS ON DAY 307,1979
ILLEGAL RECORD AT 2230 HOURS ON DAY 307,1979:
//////
1979 307 2230 CHANNEL 17 : UNEXPECTED CHARACTER : [illegible]
END OF DATA SET FOR 2230 HOURS ON DAY 307,1979 :    [illegible] CHANNELS READ
ERROR DURING SEARCH: 1979 307 2230
X001979
1979 307 2300 CHANNEL 22 : SENSOR IN RANGE
1979 307 2300 CHANNEL 24 : SENSOR IS OK
1979 307 2300 CHANNEL 23 : UPPER LIMIT EXCEEDED.
DUPLICATE DATA FOUND FOR CHANNEL 24 AT 2300 HOURS ON DAY 307,1979:
A 24 -0.3100 V
EOF DURING SEARCH: 1979 307 2300
UNSPECIFIED PROCESS FOR CHANNEL 16 (ITERATION =  1)
1979 307 2200 CHANNEL 18 : HUMIDITY OUT OF RANGE (   [illegible]); CORRECTED TO    0.
1979 307 2200 CHANNEL 21 : HUMIDITY OUT OF RANGE (  [illegible]); CORRECTED TO 100.
1979 307 2200 CHANNEL 23 : UNEXPECTED UNITS (MV) FOUND
1979 307 2230 CHANNEL 23 : CORRECT UNITS ( V) FOUND
1979 307 2230 CHANNEL 25 : W.D. VOLTAGE OUT OF RANGE ( [illegible] V)
1979 307 2230 CHANNEL 26 : POS. PWR. SUPPLY OUT OF RANGE ( 44.198 V)
1979 307 2300 CHANNEL 27 : NEG. PWR. SUPPLY OUT OF RANGE (  0.000 V)
ILLEGAL CHANNEL NUMBER 55 (ITERATION = 13)
```

Table F2

METSRT Input Errors Correlated with Error Messages

```
Y307:22:00:01
X001979
A 15 -0.0002  V
A 16 -0.0007  V
A 17 +0.7615  V
A 18<-0.0182  V
A 19 +0.0689  V
A 20 + 22.52MV
A 21 + 5.236  V
A 22 +777.77**OL
A 23$+ 45.72MV
A 24 -0.3099**BT
A 25 +3.6265  V
A 26 +16.052  V
A 27 -15.576  V
A 28 + 0.005  V
A 29 + 0.004  V
A 31 +   24.7 C
A EAC LINE
A 40 +    7.0 C
A 55 +   24.8 C
Y000:00:00:07
X0019790
A SYSTEM RESET
Y307:22:30:00
X001979
//////
A 18 +0.6350  V
A 23 -0.0002  V
A 42 -   12.4 C
A 17 +0.1234  V**
A 25 +9.6270  V
A 26 +44.198  V
A 27 -15.447  V
A 40 +    7.0 C
A 55 +   24.9 C
A 43<-   13.8 C
Y307:22:45:01
X001979
Y307:23:00:01
X001979
A 15 -0.0001  V
A 16 -0.0003  V
A 17 +0.7609  V
A 18 +0.6262  V
A 19 +0.0590  V
A 20 + 23.32MV
A 21 + 3.111  V
A 22 +2.2960  V
A 24 -0.0430  V
A 23>+0.0050  V
A 24 -0.3100  V
A 25 +3.5978  V
A 26 +16.297  V
A 27 +0.0000  V
A 28 - 0.009  V
A 29 + 0.007  V
A 31 +   24.7 C
A 40 +    7.1 C
A 55 +   24.8 C
```

```
1979 307 2200 CHANNEL 18 : LOWER LIMIT EXCEEDED.
1979 307 2200 CHANNEL 22 : SENSOR OVERLOAD
1979 307 2200 CHANNEL 23 : UNEXPECTED CHARACTER : $ + $
1979 307 2200 CHANNEL 24 : SENSOR BROKEN
INCOMPREHENSIBLE DATA AT 2200 HOURS ON DAY 307,1979:
A EAC LINE
SYSTEM RESET AFTER 2200 HOURS ON DAY 307,1979. DATA SET(S) SKIPPED.
ACQUISITION RESUMED AT 2230 HOURS ON DAY 307,1979
ILLEGAL RECORD AT 2230 HOURS ON DAY 307,1979:
//////
1979 307 2230 CHANNEL 17 : UNEXPECTED CHARACTER : * + **
END OF DATA SET FOR 2230 HOURS ON DAY 307,1979 :   3 CHANNELS READ
ERROR DURING SEARCH: 1979 307 2230
X001979
1979 307 2300 CHANNEL 22 : SENSOR IN RANGE
1979 307 2300 CHANNEL 24 : SENSOR IS OK
1979 307 2300 CHANNEL 23 : UPPER LIMIT EXCEEDED.
DUPLICATE DATA FOUND FOR CHANNEL 24 AT 2300 HOURS ON DAY 307,1979:
A 24 -0.3100 V
EOF DURING SEARCH: 1979 307 2300
UNSPECIFIED PROCESS FOR CHANNEL 16 (ITERATION =  1)
1979 307 2200 CHANNEL 18 : HUMIDITY OUT OF RANGE (   -4.) ; CORRECTED TO   0.
1979 307 2200 CHANNEL 21 : HUMIDITY OUT OF RANGE (  105.) ; CORRECTED TO 100.
1979 307 2200 CHANNEL 23 : UNEXPECTED UNITS (MV) FOUND
1979 307 2230 CHANNEL 23 : CORRECT UNITS ( V) FOUND
1979 307 2230 CHANNEL 25 : W.D. VOLTAGE OUT OF RANGE ( 9.6270 V)
1979 307 2230 CHANNEL 26 : POS. PWR. SUPPLY OUT OF RANGE ( 44.198 V)
1979 307 2300 CHANNEL 27 : NEG. PWR. SUPPLY OUT OF RANGE (  0.000 V)
ILLEGAL CHANNEL NUMBER 55 (ITERATION = 13)
```

Table F3

METSRT Processing Time Errors and the Corresponding Warning Messages

```
Y307:22:00:01
X001979
A 15 -0.0002 V
A 16 -0.0007 V
A 17 +0.7615 V
A 18<-0.0182 V
A 19 +0.0689 V
A 20 + 22.52MV
A 21 + 5.236 V
A 22 +777.77**OL
A 23$+ 45.72MV
A 24 -0.3099**BT
A 25 +3.6265 V
A 26 +16.052 V
A 27 -15.576 V
A 28 + 0.005 V
A 29 + 0.004 V
A 31 + 24.7 C
A BAD LINE
A 40 +   7.0 C
A 55 +  24.6 C
Y000:00:00:07
X0019790
A SYSTEM RESET
Y307:22:30:00
X001979
//////
A 18 +0.6350 V
A 23 -0.0002 V
A 42 -  12.4 C
A 17 +0.1234 V**
A 25 +9.6270 V
A 26 +44.198 V
A 27 -15.447 V
A 40 +  7.0  C
A 55 +  24.9 C
A 43<-  13.8 C
Y307:22:45:01
X001979
Y307:23:00:01
X001979
A 15 -0.0001 V
A 16 -0.0003 V
A 17 +0.7609 V
A 18 +0.6262 V
A 19 +0.0590 V
A 20 + 23.32MV
A 21 + 3.111 V
A 22 +2.2960 V
A 24 -0.0430 V
A 23>+0.0050 V
A 24 -0.3100 V
A 25 +3.5978 V
A 26 +16.297 V
A 27 +0.0000 V
A 28 - 0.001 V
A 29 + 0.007 V
A 31 +  24.7 C
A 40 +   7.1 C
A 55 +  24.8 C
```

```
1979 307 2200 CHANNEL 18 : LOWER LIMIT EXCEEDED.
1979 307 2200 CHANNEL 22 : SENSOR OVERLOAD
1979 307 2200 CHANNEL 23 : UNEXPECTED CHARACTER : CASE = $
1979 307 2200 CHANNEL 24 : SENSOR BROKEN
INCOMPREHENSIBLE DATA AT 2200 HOURS ON DAY 307,1979:
A BAD LINE
SYSTEM RESET AFTER 2200 HOURS ON DAY 307,1979. DATA SET(S) SKIPPED.
ACQUISITION RESUMED AT 2230 HOURS ON DAY 307,1979
ILLEGAL RECORD AT 2230 HOURS ON DAY 307,1979:
//////
1979 307 2230 CHANNEL 17 : UNEXPECTED CHARACTER : CASE = **
END OF DATA SET FOR 2230 HOURS ON DAY 307,1979 :   3 CHANNELS READ
ERROR DURING SEARCH: 1979 307 2230
X001979
1979 307 2300 CHANNEL 22 : SENSOR IN RANGE
1979 307 2300 CHANNEL 24 : SENSOR IS OK
1979 307 2300 CHANNEL 23 : UPPER LIMIT EXCEEDED.
DUPLICATE DATA FOUND FOR CHANNEL 24 AT 2300 HOURS ON DAY 307,1979:
A 24 -0.3100 V
EOF DURING SEARCH: 1979 307 2300
UNSPECIFIED PROCESS FOR CHANNEL 16 (ITERATION =  1)
1979 307 2200 CHANNEL 18 : HUMIDITY OUT OF RANGE (   -2.) ; CORRECTED TO   0.
1979 307 2200 CHANNEL 21 : HUMIDITY OUT OF RANGE (  133.) ; CORRECTED TO 100.
1979 307 2200 CHANNEL 23 : UNEXPECTED UNITS (MV) FOUND
1979 307 2230 CHANNEL 23 : CORRECT UNITS ( V) FOUND
1979 307 2230 CHANNEL 25 : W.D. VOLTAGE OUT OF RANGE ( 9.6270 V)
1979 307 2230 CHANNEL 26 : POS. PWR. SUPPLY OUT OF RANGE ( 44.198 V)
1979 307 2300 CHANNEL 27 : NEG. PWR. SUPPLY OUT OF RANGE (  0.000 V)
ILLEGAL CHANNEL NUMBER 55 (ITERATION = 13)
```

Appendix G
Non-Standard FORTRAN

An attempt has been made to restrict the statements used in these programs to the set defined by 1966 ANSI standard FORTRAN. However, in several places non-standard statements have been used. Often this was done because the desired function was sufficiently complex that no simple alternative was available. In other cases, the non-standard statements were considered to involve relatively trivial functions which could easily be deleted by other users without detriment to the overall program function.

In this appendix we briefly discuss these non-standard features and suggest possible alternatives for some of them.

I. PROGRAM 'name'

This statement assigns a name to the main program just as FUNCTION or SUBROUTINE are used to designate subprograms. It may be omitted without affecting any program functions.

II. PARAMETER M=n

PARAMETER M = n assigns, at compile time, the value 'n' to the constant 'M'. In the METEOR package, PARAMETER statements are used to set MDATA (the maximum allowable number of data sets) and MCHAN (maximum possible number of data channels). These two constants are then used to dimension many of the arrays in both the main programs and in the subprograms. A PARAMETER statement must appear in each subprogram in which MDATA or MCHAN are to be used.

If the PARAMETER statements are omitted, then each occurrence of MDATA and MCHAN must be replaced by explicit values.

III. OPEN/CLOSE

These statements control the characteristics of the files used for input and output. The following arguments may be used with OPEN or CLOSE statements:

1)	UNIT	= n	Defines the logical unit number.
2)	DEVICE	= 'DSK'	Specifies that the device is a disk.
3)	ACCESS	= 'SEQOUT'	Initializes device for write.
		= 'SEQIN'	Sets device for read.
		= 'APPEND'	Sets device for write but does not initialize. New data will be added to the end of the existing file.
4)	DISPOSE	= 'DELETE'	Delete file after it is closed.
		= 'SAVE'	Save file after close. This is the default.
5)	FILE	= 'filename'	Allows new files to be named.

The default name is FORØn.DAT, where 'n' is the logical unit number.

6) ERR = s — Causes a branch to statement number 's' if an I/O error occurs.

In many systems, the functions of the OPEN and CLOSE statements may be performed by job control commands external to the program. However, the error recovery function may not be available in these cases.

METINP closes and reopens file FORØ4.DAT at several points. These statements could be eliminated and the file allowed to remain open continuously during program execution.

In METCLC, subroutine DATOUT closes FORØ7.DAT for input, reopens it for output, and rewrites the entire file (with modifications). The equivalent effect might be achieved by defining a new logical unit to receive this output [change the WRITE (7,f) statements to WRITE (8,f), for example], deleting the old file 7, and renaming the new file (file 8, in this example).

IV. STOP 'string'

This statement causes the message 'string' to be written to the default device (TTY for interactive jobs, LOG file for batch jobs) at the time that the STOP is encountered. These statements serve little purpose in batch jobs (in most cases the same message is available in FORØ6.DAT) but have proven to be convenient in debugging from a terminal. They may be replaced by standard STOPs.

V. RETURN n

This statement allows subroutines to return to any point in the calling program. Any subroutine that uses this feature must have one or more '&s' arguments (where 's' is a statement number) in the CALL and corresponding dummy arguments ,'*', in the SUBROUTINE statement. A RETURN n will then return to the statement number represented by the n^{th} asterisk (counting from the left).

A substitute for this function might involve setting the value of a variable within the subroutine and then using a computed GO TO in the calling program to branch to the desired statement number.

VI. END = s/ERR = s

The END = s feature is used as part of the READ statement to direct the program to statement 's' if an End of File (EOF) is read. The format of the statement is [READ (n,f,END = s) 'list']. Since this is only used to enable the program to print out an appropriate message before termination of the program it is not really necessary and may be omitted.

If it is desirable to retain this feature, the function could possibly be

simulated by placing some standard character in the last record of each file. The input may then be tested for the presence of this character and an EOF routine called if it is found. This procedure might be implemented by doing a READ with an A-format, then using a BACKSPACE and another READ (with a different format) if the special character was not found.

ERR = s is also used in a READ statement in essentially the same fashion as the END function. ERR operates as described in the discussion of OPEN and CLOSE statements. Since the error, if present, is detected by the operating system's I/O routines there is little that can be done to mimic this function within the FORTRAN program. However, it is possible that the job control language may provide commands by means of which an error recovery may be accomplished.

VII. SKIP RECORD n

The SKIP RECORD statement causes the next record on device 'n' to be skipped during input. It is equivalent to the construction:

```
  READ (n,f)
f FORMAT (/)
```

VIII.ENCODE/DECODE

These two statements allow data to be reformatted within the computer. They both require arguments as follows:

```
ENCODE (n,f,'array') 'list'
DECODE (n,f,'array') 'list'
```

where 'n' is the number of characters to be transferred and 'f' is a format statement number.

ENCODE is somewhat like WRITE in that information in the variables specified by 'list' is transferred to a string under control of a FORMAT statement. However, instead of being written to an output device, the string is written into variable 'array'.

Conversely, DECODE reads 'n' characters contained in 'array', formats them as specified by FORMAT statement 'f', and stores the results in the variables given in 'list'.

ENCODE is used by METPLT to create character strings which are used as captions and axis labels. These strings could be explicitly defined in the program, read in from a file, or they may be omitted entirely without significantly altering the functions of these programs.

METINP reads a record once in an A-format, then uses DECODE to reformat the string as required. This could be accomplished by using a BACKSPACE followed by another READ.